

# Optimal Assumptions for Synthesis

Romain Brenguier (University of Oxford)

April 13, 2016

## Abstract

Controller synthesis is the process of constructing a correct system automatically from its specification. This often requires assumptions about the behaviour of the environment. It is difficult for the designer to identify the assumptions that ensures the existence of a correct controller, and doing so manually can lead to assumptions that are stronger than necessary. As a consequence the generated controllers are sub optimal in terms of generality and robustness. In this work, given a specification, we identify the weakest assumptions that ensures the existence of a controller. We also consider two important classes of assumptions: the ones that can be ensured by the environment and assumptions that speaks only about inputs of the systems. We show that optimal assumptions correspond to strongly winning strategies, admissible strategies and remorse-free strategies respectively. Based on this correspondence, we propose an algorithm for computing optimal assumptions that can be ensured by the environment.

## 1 Introduction

Reactive systems are in outgoing interaction with their environment. The goal of synthesis is the implementation of a correct reactive system from a high level specification. A specification is given by a language over the alphabet of input and output signals of the desired system or program. We will consider  $\omega$ -regular languages which are a powerful and natural way to describe the behaviour of reactive systems. A specification is realisable if for all finite sequence of input signal we can produce at the same rate a sequence of output signals, such that the resulting sequence will belong to the language. When this is the case we can implement a correct program with respect to the specification. For regular languages this can be done using finite memory and thus implemented using Moore machines.

In general, the realisation of a specification requires some assumption about the behaviour of the environment. In this work, given a specification, we compute the weakest assumption that makes it realisable. Apart from looking for any assumption we also consider two important classes of assumptions: ensurable and input assumptions. Ensurable assumptions are the one that can be ensured by the environment; in other term they cannot be falsified by a strategy of the controller. These assumptions are natural to consider in a reactive environment context. Input assumptions are independent of the sequence of output that is produced. They are better suited than more general classes when the behaviour of the environment does not depend on the outputs of our system.

Synthesis is in general achieved by computation of winning strategies in a game. For instance, if the specification is given by a parity automaton, we can see it has a game where the controller chooses output symbols and the adversary controls input symbols. The existence of a winning strategy for the controller means that there is an implementation of the system such that all executions satisfies the specification and answers the realizability question. When winning strategies do not exist, different classes have been introduced to characterise the ones that make a best-effort. In particular, strongly winning strategy [8] play a winning strategy as soon as the current history (sequence of signals seen so far) renders the existence of a winning strategy possible. Admissible strategies [1] are not *dominated* by other ones, in the sense that no strategy performs better than them against all adversary strategies. Remorse-free strategies [6] are strategies for which no other performs better than them against all *words* played by the adversary. We draw a link between classes of assumptions and these classes of strategies.

**Example** As an example, assume we want to design a sender on a network where packets can be lost or corrupted, and our goal is to obtain a protocol similar to the classical bit alternation protocol. The outputs of the sender are actions  $send_0$  and  $send_1$ , and the environment controls  $ack_0$ ,  $ack_1$  corresponding to acknowledgement of good reception of the packet. The specification of the system are described by the  $\omega$ -regular expression:  $\Sigma_I \cdot (send_0 \cdot (\neg ack_0 \cdot send_0)^* \cdot ack_0 \cdot send_1 \cdot (\neg ack_1 \cdot send_1)^* \cdot ack_1)^\omega$ . Intuitively, we have to send message with bit control 0 until receiving the corresponding acknowledgement then do the same thing with the next message with bit control 1 and repeat this forever.

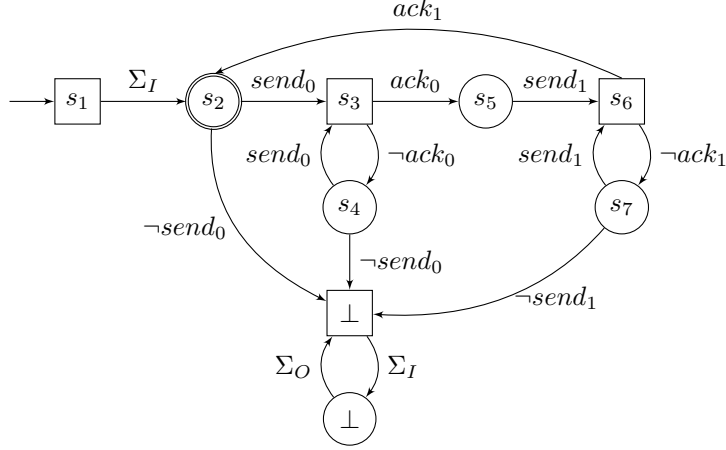


Figure 1: A specification by a Büchi automaton of the language  $\Sigma_I \cdot (send_0 \cdot (\neg ack_0 \cdot send_0)^* \cdot ack_0 \cdot send_1 \cdot (\neg ack_1 \cdot send_1)^* \cdot ack_1)^\omega$ . Accepting states (with parity colour 0) are double lined. Square states mean that the next signal is an input, while circles mean it will be an output.

Although the implementation of the protocol seems straightforward, classical realizability fails here since if all packets are lost after some point the specification will not be satisfied. To ensure realizability we have to make the assumption that a packet that is repeatedly sent will eventually be acknowledged. An admissible strategy for this specification can be implemented by a Moore machine which has the same structure as the automaton in Figure 1 with output function  $G$  such that  $G(s_2) = G(s_4) = send_0$  and  $G(s_5) = G(s_7) = send_1$ . This implementation is natural for the given specification and corresponds indeed to the bit alternation protocol. The assumption corresponding to this strategy is the language recognised by the same automaton where we add  $\perp$ -states to the set of accepting states. As we will see in Thm. 25, it is an optimal ensurable assumption for the specification.

**Scenarios** Giving specifications for our system is a way to disallow some behaviours that are not desirable. Dually, we may want to specify execution scenarios that should be possible in the synthesised system. We ask then for a system whose outcomes are all within the specifications and contains all the given scenarios.

**Generalisation** Sometimes, some particular assumptions are natural for the problem we consider but we want to synthesise a system which is as robust as possible by generalising this assumption. For instance, for the bit alternation protocol we could suggest as an initial assumption that two successive packets cannot be lost. The synthesised systems would offer no guarantee two packets in a row are lost. By generalising the assumption, we ensure that the strategy synthesised works well for the initial assumptions we have in mind, and for as many input sequences as possible. For the bit alternation protocol the protocol works well if not all packets are lost after some point in time.

**Contribution** In this article we establish correspondences between class of assumptions and classical classes of strategies. These correspondences are summarised in the following table.

Class of assumption:	Optimal achieved by:	
General ( $\mathcal{A}$ )	strongly winning strategies	Thm. 13
Ensurable ( $\mathcal{E}$ )	admissible strategies	Thm. 25
Input ( $\mathcal{I}$ )	remorsefree strategies	Thm. 46

Based on these results, we show existence of optimal assumptions in most case and give algorithms to compute optimal assumptions. In particular, we show:

- Existence of sufficient input assumptions compatible with a scenario is always true (Thm. 7). It is also true for safety assumption if the scenario is itself a safety language (Thm. 8).

- There may exist an infinite number of optimal and ensurable-optimal assumptions (Thm. 15) and of input-optimal assumptions (Thm. 41).
- We can compute an optimal ensurable assumption in exponential time for parity specification and in polynomial time if we have an oracle to solve parity games (Thm. 34 and Thm. 37).
- There is an exponential algorithm that given a sufficient assumption  $H$ , generalises it by finding  $H'$  such that  $H \subseteq H'$  and  $H'$  is ensurable-optimal (Thm. 40).

**Comparison with previous works on assumptions for synthesis** In [5], the study is focused on safety conditions defined by forbidding edges of the automaton defining specification  $L$ . This approach is less general than ours since it depends on the choice of the automaton representing  $L$ .

In the setting of [5], comparison between assumptions is based on the number of edges, while we compare them based on language inclusion which we find more relevant. Consider the example of Figure 2 taken from [5]. Player 1 has no winning strategy from  $s_1$ . According to [5], there are 2 minimal sufficient assumptions which are  $E'_s = \{(s_3, s_6)\}$  and  $E'_s = \{(s_5, s_7)\}$ . However if we remove the edge from  $s_3$  to  $s_6$ ,  $s_5$  is no longer accessible which means that the first assumption is in fact stronger than the second one, taking the point of view of language inclusion.

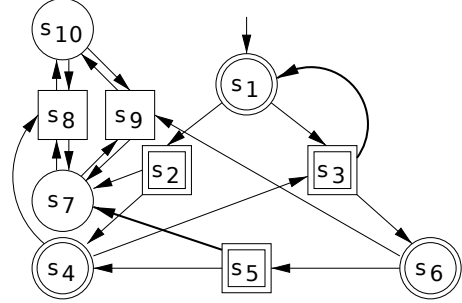


Figure 2: A game from [5].

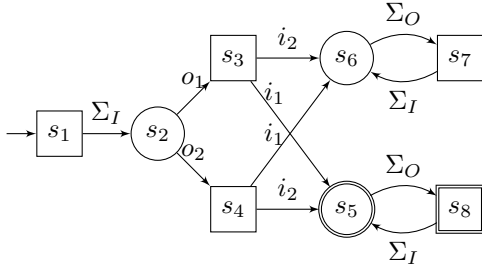


Figure 3: Automaton for specification  $\Sigma_I \cdot (o_1 \cdot i_1 \mid o_2 \cdot i_2) \cdot \Sigma_O \cdot (\Sigma_I \cdot \Sigma_O)^\omega$ .

Moreover, we show that even for this restricted class of safety assumption, the claim that there is a unique optimal assumption ([5, Thm. 5]) is wrong. Consider the example of Figure 3. Removing  $(s_3, s_6)$  or  $(s_4, s_5)$  is sufficient for  $L = \Sigma_I \cdot (o_1 \cdot i_1 \mid o_2 \cdot i_2) \cdot \Sigma_O \cdot (\Sigma_I \cdot \Sigma_O)^\omega$ . Note that both these assumptions are ensurable since they lead to immediate violation of the safety objective. This therefore contradicts [5, Thm. 5] which claims that if  $L \neq \emptyset$  then there exists a unique minimal safety assumption that is ensurable and sufficient for  $L$ .

## 2 Preliminaries

Given a finite alphabet  $\Sigma$  and an infinite word  $w \in \Sigma^\omega$ , we use  $w_i$  to denote the  $i$ -th symbol of  $w$ , and  $w_{\leq i} = w_1 \cdots w_i$  the finite prefix of  $w$  of length  $i$ . We write  $|w_{\leq i}| = i$  its length.

### 2.1 Classical realizability

A reactive system reads *input signals* in a finite alphabet  $\Sigma_I$  and produces *output signals* in a finite alphabet  $\Sigma_O$ . We assume for the rest of this paper that these alphabets are fixed. A *specification* of a reactive system is an  $\omega$ -regular language  $L \subseteq (\Sigma_I \cdot \Sigma_O)^\omega$ . A *program* or *strategy* is a mapping  $\sigma_\exists: (\Sigma_I \cdot \Sigma_O)^* \cdot \Sigma_I \mapsto \Sigma_O$ . An *outcome* of such a strategy  $\sigma_\exists$  is a word  $w$  such that for all  $i \in \mathbb{N}$ ,  $w_{2 \cdot i + 2} = \sigma_\exists(w_{\leq 2 \cdot i + 1})$ . We write  $\text{Out}(\sigma_\exists)$  for the set of outcomes of  $\sigma_\exists$ .

**Realizability problem [12]** Given a specification  $L$ , the realizability problem asks whether there exists a strategy  $\sigma_\exists$  such that  $\text{Out}(\sigma_\exists) \subseteq L$ . Such a strategy is said *winning* for  $L$ . The process of constructing such a strategy is called *synthesis*.

**Parity automata** We will assume that specifications are given by deterministic parity automata, which can recognise any  $\omega$ -regular languages [11]. A *parity automaton* is given by  $\langle S, s_0, \Delta, \chi \rangle$ , where  $S$  is a finite set of states,  $s_0 \in S$  is the initial state,  $\Delta \in S \times (\Sigma_I \cup \Sigma_O) \times S$  is the transition relation, and  $\chi: S \rightarrow \mathbb{N}$  is a colouring function. A path  $\rho \in S^\omega$  is accepting if the smallest colour seen infinitely often is even (i.e. if  $\min\{c \mid \forall i \in \mathbb{N}. \exists j \geq i. \chi(w_j) = c\} \in 2 \cdot \mathbb{N}$ ). A word  $w$  is accepted if there is an accepting path whose labelling is  $w$ . A *Büchi automaton* is a parity automaton for which  $\chi(S) \subseteq \{0, 1\}$ . A *safety automaton* is a Büchi automaton where states of colour 1 are absorbing. The language recognised by an automaton is the set of words it accepts. In practice specification are often given in temporal logics such as LTL before being translated to an automaton representation. In our examples, we will sometimes mention LTL formulas, using the syntax  $\mathbf{X}\phi$  meaning  $\phi$  holds in the next state,  $\phi_1 \mathbf{U} \phi_2$  meaning  $\phi_1$  holds until  $\phi_2$  holds (and  $\phi_2$  must hold at some point),  $\mathbf{F}\phi := \mathbf{true} \mathbf{U} \phi$  and  $\mathbf{G}\phi := \neg \mathbf{F}(\neg \phi)$ .

**Strategies** The realizability problem is best seen as a game between two players [10]. The environment chooses the input signals and the controller the output signals. We therefore also define the concept of *environment-strategy* which is a mapping  $\sigma_\forall: (\Sigma_I \cdot \Sigma_O)^* \mapsto \Sigma_I$ . Given an environment-strategy  $\sigma_\forall$ , we write  $\text{Out}(\sigma_\forall)$  the word  $w$  such that for all  $i \in \mathbb{N}$ ,  $w_{2 \cdot i + 1} = \sigma_\forall(w_{\leq 2 \cdot i})$ . Given an input word  $u \in \Sigma_I^\omega$ , we write  $\text{Out}(\sigma_\exists, u)$  the unique outcome such that for all  $i \in \mathbb{N}$ ,  $w_{2 \cdot i + 1} = u_{i+1}$  and  $w_{2 \cdot i + 2} = \sigma_\exists(w_{\leq 2 \cdot i + 1})$ . We also write  $\text{Out}(\sigma_\exists, \sigma_\forall) = \text{Out}(\sigma_\exists) \cap \text{Out}(\sigma_\forall)$ , note that it contains only one outcome. A finite prefix of an outcome is called a *history*. Given a history  $h$ , we write  $\text{Out}_h(\sigma_\exists)$  a word  $w$  such that for all  $i \leq |h|$ ,  $w_i = h_i$  and for all  $i$  such that  $2 \cdot i + 2 > |h|$ ,  $w_{2 \cdot i + 2} = \sigma_\exists(w_{\leq 2 \cdot i + 1})$ .

We write  $\pi_I$  and  $\pi_O$  the *samplings* over input and output signals respectively, that is  $\pi_I: (\Sigma_I \cdot \Sigma_O)^\omega \rightarrow \Sigma_I^\omega$  is such that  $\pi_I(w)_i = w_{2 \cdot i - 1}$  and  $\pi_O: (\Sigma_I \cdot \Sigma_O)^\omega \rightarrow \Sigma_O^\omega$  is such that  $\pi_O(w)_i = w_{2 \cdot i}$ .

**Moore machine** In practice strategies are implemented using Moore machines, that correspond to strategies that only use finite memory. A *Moore machine* is given by  $\langle S_I, S_O, s_0, \delta, G \rangle$  where  $S = S_I \cup S_O$  is a finite set of states,  $S_I$  is a set of input states and  $S_O$  of output states,  $s_0 \in S_I$  is the initial state,  $\delta \in S \times \Sigma_I \cup \Sigma_O \rightarrow S$  is the transition function, and  $G: S_O \rightarrow \Sigma_O$  is an output function. A Moore machine implements a strategy  $\sigma_\exists$  where for all history  $h \in (\Sigma_I \cdot \Sigma_O)^* \cdot \Sigma_I$ ,  $\sigma_\exists(h) = G(s_{|h|})$  where for all  $0 \leq i < |h|$ ,  $s_{i+1} = \delta(s_i, h_{i+1})$ . Note that our Moore machine read both inputs and outputs. In many application there would be no need to read the outputs since it can be left undefined on histories that are incompatible with the strategy. However we prefer this definition which is coherent with our definition of strategies and makes it easier to combine strategies which may not be compatible with the same histories.

**Example 1.** In all the examples of this article we will assume that the set of input signals is  $\Sigma_I = \{i_1, i_2\}$  and the set output of output signals is  $\Sigma_O = \{o_1, o_2\}$ . The specification given in Figure 4 is realisable. The corresponding winning strategy consists for the controller to invert the input signals: if  $i_1$  is the input then in next round we do not output  $o_1$  and vice-versa. Formally, this strategy is given for all history  $h \in (\Sigma_I \cdot \Sigma_O) \cdot \Sigma_I$  by if  $h_{|h|} = i_1$  then  $\sigma_\exists(h) = o_2$  and if  $h_{|h|} = i_2$  then  $\sigma_\exists(h) = o_1$ . A Moore machine which implement a winning strategy can also be obtained from the parity automaton by setting the output function to be such that  $G(s_2) = o_2$ ,  $G(s_1) = o_1$ .

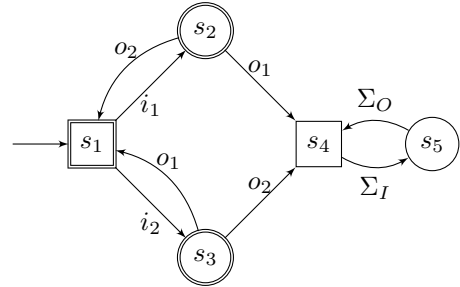


Figure 4: A specification by a Büchi automaton corresponding to LTL formula  $\mathbf{G}(i_1 \Leftrightarrow \mathbf{X}(\neg o_1))$ . Accepting states (with parity colour 0) are double lined. Square states mean that the next signal is an input, while circles mean it will be an output.

## 2.2 Assumptions

An *assumption* is an  $\omega$ -regular language  $A \subseteq (\Sigma_I \cdot \Sigma_O)^\omega$ . Assumption  $A$  is *sufficient* for specification  $L$  if there is a strategy  $\sigma_\exists$  of the controller such that any outcome either satisfies  $L$  or is not in  $A$ , i.e.

$\text{Out}(\sigma_{\exists}) \cap A \subseteq L$ . In that case we also say that  $A$  is *sufficient for*  $\sigma_{\exists}$ . We are looking for assumptions that are the least restrictive. We say that assumption  $A$  is *less restrictive* than  $B$  if  $A \subseteq B$ . We say it is *strictly less restrictive* if  $A \subset B$  (i.e.  $A \subseteq B$  and  $A \neq B$ ).

We will consider several class of assumptions:

- The general class of assumptions is written  $\mathcal{A} = (\Sigma_I \cdot \Sigma_O)^\omega$ .
- An *input-assumption* is an assumption which concerns only inputs and does not restrict at all outputs.  $A$  is an input-assumption if for all words  $w$  and  $w'$ ,  $\pi_I(w) = \pi_I(w') \wedge w \in A \Rightarrow w' \in A$ . We write this class  $\mathcal{I} = \{A \in \mathcal{A} \mid \forall w, w' \in (\Sigma_I \cdot \Sigma_O)^\omega. \pi_I(w) = \pi_I(w') \wedge w \in A \Rightarrow w' \in A\}$ .
- We say that  $A$  is *ensurable*, if the environment has a winning strategy for  $A$ , i.e. for each  $w \in (\Sigma_I \cdot \Sigma_O)^*$ , if  $w \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap A \neq \emptyset$  then there exists an environment strategy  $\sigma_{\forall}$  such that  $w \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap \text{Out}(\sigma_{\forall}) \neq \emptyset$  and  $\text{Out}(\sigma_{\forall}) \subseteq A$ . We write  $\mathcal{E}$  for this class. The fact that the environment can ensure the assumption is often a requirement done in synthesis (see for instance [2]).
- We say that an assumption  $A$  is *output-restrictive* if it restricts the strategies of the controller, that is if there is  $w \in (\Sigma_I \cdot \Sigma_O)^*$  and  $\sigma_{\exists}$  a strategy, such that

$$\left\{ \begin{array}{l} A \cap w \cdot (\Sigma_I \cdot \Sigma_O)^\omega \neq \emptyset \\ \text{and } \text{Out}(\sigma_{\exists}) \cap w \cdot (\Sigma_I \cdot \Sigma_O)^\omega \neq \emptyset \\ \text{and } A \cap \text{Out}(\sigma_{\exists}) \cap w \cdot (\Sigma_I \cdot \Sigma_O)^\omega = \emptyset \end{array} \right.$$

Intuitively an output-restrictive assumption  $A$  forbids strategy  $\sigma_{\exists}$ , so playing  $\sigma_{\exists}$  would be a trivial way to satisfy  $A \Rightarrow L$ . From the point of view of synthesis it is better to avoid such assumptions. We write this class  $\mathcal{O}$ .

- Given an assumption  $A$ , we write the set of bad prefixes  $\text{Bad}(A) = \{h \in (\Sigma_I \cdot \Sigma_O)^* \mid h \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap A = \emptyset\}$ . Assumption  $A$  is a *safety assumption* if every word not in  $A$  has a bad prefix [9], i.e.  $A = (\Sigma_I \cdot \Sigma_O)^\omega \setminus \{w \mid \exists k. w_{\leq 2 \cdot k} \in \text{Bad}(A)\}$ . We write  $\mathcal{S}$  for this class.

For a class  $\mathcal{C}$  of assumptions, we say that assumption  $A$  is  $\mathcal{C}$ -*optimal* for  $L$  if  $A$  belongs to  $\mathcal{C}$ , is sufficient for  $L$  and there is no assumption  $B \in \mathcal{C}$  that is strictly less restrictive than  $A$  and sufficient for  $L$ .

*Remark.* Note that  $L$  is always a sufficient assumptions, however it is too strong and will never be interesting for synthesis: if we assume that our specification always hold then there is nothing left to do. That is why we ask for assumptions that are as weak as possible.

We now prove the relationships that exist between the classes of assumptions.

**Lemma 2.** *Non-empty input assumptions are ensurable, i.e.  $(\mathcal{I} \setminus \{\emptyset\}) \subset \mathcal{E}$*

*Proof.*  $(\mathcal{I} \setminus \{\emptyset\}) \subseteq \mathcal{E}$ : Let  $A$  be a non-empty input-assumption, and  $w \in A$ . Let  $\sigma_{\forall}$  be the strategy of the environment that follows  $w$ , that is for all histories  $h$ ,  $\sigma_{\forall}(h) = w_{|h|+1}$ . For all outcome  $u \in \text{Out}(\sigma_{\forall})$ ,  $\pi_I(u) = \pi_I(w)$ . Since  $\pi_I(w) = \pi_I(u)$  and  $A$  is an input assumption  $u \in A$ . Hence  $A$  is ensurable.

$(\mathcal{I} \setminus \{\emptyset\}) \neq \mathcal{E}$ : Consider the specification  $L = \text{X}o_1 \Leftrightarrow \text{X}X i_1$ , it is ensurable by any strategy such that  $\sigma_{\forall}(\Sigma_I \cdot o_1) = i_1$  and  $\sigma_{\forall}(\Sigma_I \cdot o_2) = i_2$ . However it is not an input-assumption:  $w = (i_1 \cdot o_1)^\omega \in L$  and  $w' = (i_1 \cdot o_2)^\omega \notin L$  but  $\pi_I(w) = \pi_I(w')$ .  $\square$

**Lemma 3.** *For  $\omega$ -regular specifications, ensurable assumptions are exactly the assumptions that are not output-restrictive,  $\mathcal{E} = \mathcal{G} \setminus \mathcal{O}$ .*

*Proof.*  $\mathcal{E} \subseteq \mathcal{G} \setminus \mathcal{O}$ : Let  $A$  be an ensurable assumption, and  $w \in (\Sigma_I \cdot \Sigma_O)^*$ . Assume there is a strategy  $\sigma_{\exists}$  such that  $A \cap w \cdot (\Sigma_I \cdot \Sigma_O)^\omega \neq \emptyset$  and  $\text{Out}(\sigma_{\exists}) \cap w \cdot (\Sigma_I \cdot \Sigma_O)^\omega \neq \emptyset$ , we prove that  $A \cap \text{Out}(\sigma_{\exists}) \cap w \cdot (\Sigma_I \cdot \Sigma_O)^\omega \neq \emptyset$ , which shows that  $A$  is nonrestrictive. Let  $u \in A \cap w \cdot (\Sigma_I \cdot \Sigma_O)^\omega$  by the definition of ensurable assumptions there exists  $\sigma_{\forall}$  compatible with  $w$  and such that  $\text{Out}(\sigma_{\forall}) \cap A$ . Let  $w' = \text{Out}(\sigma_{\exists}, \sigma_{\forall})$ ,  $w$  is a prefix of  $w'$  since

it is compatible with both  $\sigma_{\exists}$  and  $\sigma_{\forall}$ . Since  $\text{Out}(\sigma_{\forall}) \cap A$ ,  $w' \in A$  so  $w \in A \cap \text{Out}(\sigma_{\exists}) \cap w \cdot (\Sigma_I \cdot \Sigma_O)^\omega$ . Hence  $A$  is nonrestrictive.

$\mathcal{G} \setminus \mathcal{O} \subseteq \mathcal{E}$ : Let  $A$  be an assumption which is not output restrictive. Assume towards a contradiction that there is some history  $w$  such that  $w \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap A \neq \emptyset$  and environment has no winning strategy  $\sigma_{\forall}$  for  $A$  which is compatible with  $w$ . This means environment has no winning strategy for  $w^{-1} \cdot A = \{w' \in (\Sigma_I \cdot \Sigma_O)^\omega \mid w \cdot w' \in A\}$ . By determinacy of  $\omega$ -regular objectives [11, Corollary 2.10] there is a winning strategy  $\sigma_{\exists}$  for  $(\Sigma_I \cdot \Sigma_O)^\omega \setminus (w^{-1} \cdot A)$ . Consider the strategy  $\sigma'_{\exists}$  that plays according to  $w$  and then switches to  $\sigma_{\exists}$ , that is for a history  $h$  if  $|h| < |w|$  then  $\sigma'_{\exists}(h) = w_{|h|+1}$  and if  $|h| \geq |w|$ ,  $\sigma'_{\exists}(h) = \sigma_{\exists}(h_{\geq |w|})$ . The history  $w$  is compatible with  $\sigma'_{\exists}$  so  $\text{Out}(\sigma'_{\exists}) \cap w \cdot (\Sigma_I \cdot \Sigma_O)^\omega \neq \emptyset$ . But since  $\sigma_{\exists}$  is winning for  $(\Sigma_I \cdot \Sigma_O)^\omega \setminus (w^{-1} \cdot A)$ , no outcome is of the form  $w \cdot w'$  with  $w' \in w^{-1} \cdot A$ . This means that  $A \cap \text{Out}(\sigma'_{\exists}) \cap w \cdot (\Sigma_I \cdot \Sigma_O)^\omega = \emptyset$  which contradicts the fact that  $A$  is not output restrictive.  $\square$

**Example 4.** The specification  $L = o_1 \mathbf{U} i_1$  is not realisable, however several assumptions can be sufficient for it. The automaton corresponding to  $L$  is represented in Figure 5. Consider for instance the assumption  $A$  given by the LTL formula  $F o_1$ . It is sufficient for  $L$  and is in fact sufficient for any specification since a strategy  $\sigma_{\exists}$  of Eve which never plays  $o_1$  has no outcome in  $A$ . To avoid this degenerate assumptions we focus on non-restrictive assumptions:  $F o_1$  is indeed output restrictive. On the other hand  $F(o_1) \Leftrightarrow F(i_1)$  is ensurable because the environment can react to make the assumption hold, no matter the strategy  $\sigma_{\exists}$  we chose. We can also check that it is sufficient for  $L$ : the strategy that always play  $o_1$  is winning. This is a fine assumption in the context of a reactive environment, but if the behaviour of the environment is independent of the output of the system, talking about  $o_1$  in the assumption is not relevant. In that case we would prefer for instance the input-assumption  $F i_1$  which is sufficient for  $L$  and is independent of outputs.

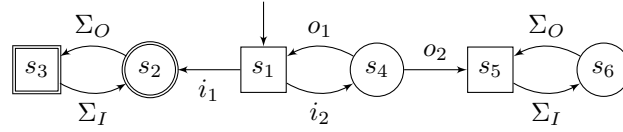


Figure 5: Büchi automaton recognising the language corresponding to LTL formula  $o_1 \mathbf{U} i_1$ .

## 2.3 Refinement using scenarios

As we will see in the next sections, in general there are an infinite number of incomparable optimal assumptions. This brings the problem of choosing one among all the possibilities. How can the algorithm know which assumption will satisfy the designer? A solution is to get some feedback from the user in the form of *scenarios*. A scenario is a behaviour that the environment can exhibit in practice and that the strategy produced in the end should allow. We will use scenarios provided by the user to chose a solution.

**Scenario** Formally a *scenario* is given by a language  $S \subseteq (\Sigma_I \cdot \Sigma_O)^\omega$ . A strategy  $\sigma_{\exists}$  is *compatible* with the set of scenarios  $S$  when  $S \subseteq \text{Out}(\sigma_{\exists})$ . Similarly  $S$  is *compatible* with specification  $L$  if  $S \subseteq L$ . A scenario  $S$  is *coherent* if there is no history  $h \in (\Sigma_I \cdot \Sigma_O)^* \cdot \Sigma_I$  prefixes of two words  $w, w' \in S$  such that  $w_{|h|+1} \neq w'_{|h|+1}$ . We say that assumption  $H$  is sufficient for  $L$  and  $S$ , if there exists  $\sigma_{\exists}$  such that  $S \subseteq H \cap \text{Out}(\sigma_{\exists}) \subseteq L$ .

If  $S$  is not coherent then a compatible strategy in  $h$  would need to allow both  $w_{|h|+1}$  and  $w'_{|h|+1}$  which is impossible. Being coherent is in fact a necessary and sufficient condition for the existence of a compatible strategy.

**Lemma 5.** *Scenario  $S$  is coherent if, and only if, there exists a strategy compatible with  $S$ . In particular, given a coherent scenario  $S$  and a strategy  $\sigma_{\exists}$ , the strategy  $[S \rightarrow \sigma_{\exists}]$  defined by:*

$$[S \rightarrow \sigma_{\exists}](h) = \begin{cases} w_{h+1} & \text{if there is } w \in S \text{ such that } h \text{ is a prefix of } w \\ \sigma_{\exists}(h) & \text{otherwise} \end{cases}$$

*is compatible with  $S$ .*

*Proof.* Let  $S$  be a coherent scenario, we show that for any strategy  $\sigma_{\exists}$ ,  $[S \rightarrow \sigma_{\exists}]$  is compatible with  $S$ . Let  $w \in S$ , and  $i \in \mathbb{N}$ . The history  $w_{2 \cdot i + 1}$  is prefix of a scenario and therefore there is  $w' \in S$  such that  $[S \rightarrow \sigma_{\exists}](w_{2 \cdot i + 1}) = w'_{2 \cdot i + 2}$ . Since  $S$  is coherent  $w_{2 \cdot i + 2} = w'_{2 \cdot i + 2}$  and  $[S \rightarrow \sigma_{\exists}]$  is compatible with  $w_{\leq 2 \cdot i + 2}$ . This shows  $w$  is compatible with  $[S \rightarrow \sigma_{\exists}]$ .

Now we will show that if  $S$  is compatible with some strategy  $\sigma_{\exists}$  then  $S$  is coherent. Let  $h$  be a prefix of some words  $w, w' \in S$ . Since  $S$  is compatible with  $\sigma_{\exists}$ ,  $w_{|h|+1} = \sigma_{\exists}(h) = w'_{|h|+1}$ . Hence  $S$  is coherent.  $\square$

**Example 6.** Consider the example in Figure 6. There are a lot of different possible assumptions we could chose from. However if we give the scenario  $\Sigma_I \cdot o_2 \cdot i_2 \cdot \Sigma_O \cdot (\Sigma_I \cdot \Sigma_O)^\omega$  then the only optimal assumption is  $(\Sigma_I \cdot o_2 \cdot i_2 + \Sigma_I \cdot o_1 \cdot \Sigma_I) \cdot \Sigma_O \cdot (\Sigma_I \cdot \Sigma_O)^\omega$ . The corresponding winning strategy consists in playing  $o_2$  for the first output.

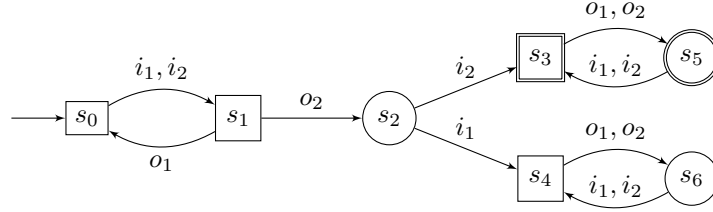


Figure 6: Büchi automaton for specification  $(\neg o_2)U(o_2 \wedge Xi_2)$ .

We say that an assumption  $A$  is  $\mathcal{C}$ -optimal for  $L$  and  $S$  if it is sufficient for  $L$  and  $S$  and there is no  $A' \in \mathcal{C}$  strictly less restrictive than  $A$  and sufficient for  $L$  and  $S$ .

## 2.4 Existence of a sufficient assumption with scenario

We show that there always exist an input assumption which is sufficient and compatible with a given scenario.

**Theorem 7.** *Let  $L$  be a specification and  $S$  a scenario. If  $S$  is compatible with  $L$ , then there exists a input assumption which is sufficient for  $L$  and compatible with  $S$ .*

*Proof.* Assume  $S$  is compatible with  $L$  and let  $H = \{w \mid \exists w' \in S. \pi_I(w) = \pi_I(w')\}$ . It is clear that  $H$  is an input assumption. We will prove that for any strategy  $\sigma_{\exists}$ ,  $H$  is sufficient for  $[S \rightarrow \sigma_{\exists}]$ . Let  $w \in H \cap \text{Out}([S \rightarrow \sigma_{\exists}])$ . Since  $w \in H$ , there is  $w' \in S$  such that  $\pi_I(w') = \pi_I(w)$ . Since  $[S \rightarrow \sigma_{\exists}]$  is compatible with  $S$  (Lem. 5), it is compatible with  $w'$ , and therefore  $w' = \text{Out}(\sigma_{\exists}, \pi_I(w)) = w$ . This proves that  $w \in S$  and therefore  $H$  is sufficient for  $L$ .  $\square$

**Theorem 8.** *Let  $L$  be a specification and  $S$  a scenario compatible with  $L$ . If  $S$  is a safety language then there exists a safety assumption which is sufficient for  $L$  and compatible with  $S$ .*

*Proof.* The set of bad prefixes of  $S$  define a safety language  $A = (\Sigma_I \cdot \Sigma_O)^\omega \setminus \{w \mid \exists k. w_{\leq 2 \cdot k} \in \text{Bad}(S)\}$ . Let  $\sigma_{\exists}$  be a strategy, and  $w \in A \cap \text{Out}([S \rightarrow \sigma_{\exists}])$ . There is no  $i$  such that  $w_{\leq i} \in \text{Bad}(S)$ . Therefore for all  $i$ ,  $w_{\leq i}$  is the prefix of some  $w' \in S$ . Hence for all  $i$ ,  $w_{\leq i}$  is not in  $\text{Bad}(S)$ , and since  $S$  is a safety language this means that  $w \in S$  and since  $S$  is compatible with  $L$ ,  $w \in L$ . Thus  $A$  is sufficient for  $L$ .  $\square$

*Remark.* There are example of scenarios that are not safety languages for which there is no sufficient safety assumption. Consider  $S$  given by  $Fi_1$  and  $L = Fi_1$ . A safety assumption  $H$  compatible with  $S$  has to contain  $Fi_1$ . Assume towards a contradiction that there is  $w \notin H$ . Since  $H$  is a safety assumption,  $w$  has a bad prefix, that is there is  $i$  such that  $w_{\leq 2 \cdot i} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap H = \emptyset$ . As  $w_{\leq 2 \cdot i} \cdot (i_1 \cdot o_1)^\omega \in S$ , this is contradiction with the fact that  $H$  is compatible with  $S$ . Therefore  $H = (\Sigma_I \cdot \Sigma_O)^\omega$  and this is not sufficient for  $L$ .

### 3 General assumptions

In this section we study general assumptions, without distinguishing the ones that are restrictive. Properties established in this section will be useful when studying ensurable assumptions.

#### 3.1 Necessary and sufficient assumptions for a strategy

Given a specification  $L$  and a strategy  $\sigma_\exists$ , we say that an assumption  $A$  is *necessary* for  $\sigma_\exists$  if  $B$  sufficient for  $\sigma_\exists$  implies  $B \subseteq A$ .

**Lemma 9.** *Given a strategy  $\sigma_\exists$  of Eve, the assumption  $EA(\sigma_\exists) = L \cup ((\Sigma_I \cdot \Sigma_O)^\omega \setminus \text{Out}(\sigma_\exists))$  is sufficient and necessary for  $\sigma_\exists$ .*

*Proof.* Let us prove that  $EA(\sigma_\exists)$  is sufficient for  $\sigma_\exists$ . Let  $w \in \text{Out}(\sigma_\exists)$ , either  $w$  is winning for condition  $L$ , or it belongs to  $\text{Out}(\sigma_\exists) \setminus L$ . In this second case, it does not satisfy assumption  $EA(\sigma_\exists)$  which is thus sufficient for  $\sigma_\exists$ .

Let us now assume that some assumption  $A$  is sufficient for  $\sigma_\exists$  and prove that then  $A \subseteq EA(\sigma_\exists)$ , which shows  $EA(\sigma_\exists)$  is necessary. Let  $w \in A$ . If  $w \in \text{Out}(\sigma_\exists)$  then since  $A$  is sufficient for  $\sigma_\exists$  we must have that  $w \models L$  and therefore  $w \in EA(\sigma_\exists)$  by definition of  $EA$ . Otherwise  $w \notin \text{Out}(\sigma_\exists)$ , then by definition of  $EA$ ,  $w \in EA(\sigma_\exists)$ .  $\square$

**Corollary 10.** *If  $A$  is optimal then there exists  $\sigma_\exists$  such that  $A = EA(\sigma_\exists)$ .*

*Proof.* Since  $A$  is sufficient, there is  $\sigma_\exists$  such that  $A$  is sufficient for  $\sigma_\exists$ . Since  $EA(\sigma_\exists)$  is necessary for  $\sigma_\exists$ , then  $A \subseteq EA(\sigma_\exists)$ . Since  $A$  is optimal and  $EA(\sigma_\exists)$  is sufficient, we also have  $EA(\sigma_\exists) \subseteq A$ . Hence the equality.  $\square$

#### 3.2 Link with strongly winning strategies

The goal of this section is to establish a link with the notion of strongly winning strategy. Intuitively this corresponds to the strategies that play a winning strategy whenever it is possible from the current history.

**Definition 11** ([8, 4]). Strategy  $\sigma_\exists$  is *strongly winning* when for all history  $h$ , if there exists  $\sigma'_\exists$  such that  $\emptyset \neq \text{Out}(\sigma'_\exists) \cap h \cdot (\Sigma_I \cdot \Sigma_O)^\omega \subseteq L$  then  $\text{Out}(\sigma_\exists) \cap h \cdot (\Sigma_I \cdot \Sigma_O)^\omega \subseteq L$ .

We will also use the notion of *subgame winning* strategies (called subgame perfect in [8]), which are such that for all history  $h$ , if there exists  $\sigma'_\exists$  such that  $\text{Out}_h(\sigma'_\exists) \subseteq L$  then  $\text{Out}_h(\sigma_\exists) \subseteq L$ .

**Lemma 12** ([8, Lem. 1]). *For every specification, there exists strongly winning and subgame winning strategies.*

**Theorem 13.** *Let  $EA(\sigma_\exists) = L \cup ((\Sigma_I \cdot \Sigma_O)^\omega \setminus \text{Out}(\sigma_\exists))$ . If strategy  $\sigma_\exists$  is strongly winning for  $L$ , then  $EA(\sigma_\exists)$  is an optimal assumption for  $L$ . Reciprocally, if  $A$  is an optimal assumption for  $L$ , then there is a strongly winning strategy  $\sigma_\exists$  such that  $A = EA(\sigma_\exists)$ .*

*Proof.*  $\Rightarrow$  First notice that by Lem. 9,  $EA(\sigma_\exists)$  is sufficient for  $\sigma_\exists$  and thus sufficient for  $L$ . Let  $A$  be an assumption which is sufficient for  $L$ , we will prove that  $EA(\sigma_\exists) \not\subseteq A$ , which shows that  $EA(\sigma_\exists)$  is optimal. Let  $\sigma'_\exists$  be a strategy for which  $A$  is sufficient. If  $A \setminus EA(\sigma_\exists) = \emptyset$  then  $A \subseteq EA(\sigma_\exists)$  which shows the property. Otherwise there exists  $w \in A \setminus EA(\sigma_\exists)$ . Since  $w \notin EA(\sigma_\exists)$  and  $L \subseteq EA(\sigma_\exists)$ ,  $w \notin L$ , i.e.  $w$  is losing. Since  $w \notin EA(\sigma_\exists)$  and  $(\Sigma_I \cdot \Sigma_O)^\omega \setminus \text{Out}(\sigma_\exists) \subseteq EA(\sigma_\exists)$ ,  $w \in \text{Out}(\sigma_\exists)$ , i.e. it is an outcome of  $\sigma_\exists$ . Since  $A \cap \text{Out}(\sigma'_\exists) \subseteq L$  and  $w \in A \setminus L$ ,  $w \notin \text{Out}(\sigma'_\exists)$ , i.e. it is not an outcome of  $\sigma'_\exists$ .

Let  $w_{\leq k}$  be the longest prefix of  $w$  that is compatible with  $\sigma'_\exists$ . Since  $\sigma_\exists$  is strongly winning and  $w$  is an outcome of  $\sigma_\exists$  which is losing, for all strategies  $\sigma''_\exists$ , either  $w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap \text{Out}(\sigma''_\exists) = \emptyset$  or  $w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap \text{Out}(\sigma''_\exists) \setminus L \neq \emptyset$ . Since  $w_{\leq k}$  is compatible with  $\sigma'_\exists$ ,  $w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap \text{Out}(\sigma'_\exists) \neq \emptyset$  and therefore there is an outcome  $w'$  of  $\sigma'_\exists$  which is losing. Since  $A$  is sufficient for  $\sigma'_\exists$ ,  $w' \notin A$ . Note that  $w'$  is not an outcome of  $\sigma_\exists$ :  $w'_{k+1} = \sigma'_\exists(w_{\leq k}) \neq \sigma_\exists(w_{\leq k})$ . Hence,  $w \in (\Sigma_I \cdot \Sigma_O)^\omega \setminus \text{Out}(\sigma_\exists) \subseteq EA(\sigma_\exists)$ . Therefore  $w' \in EA(\sigma_\exists) \setminus A$  which proves  $EA(\sigma_\exists) \not\subseteq A$ .



⊆ Let  $A$  be an optimal assumption for  $L$  and  $\sigma_{\exists}$  a strategy for which  $A$  is sufficient. Note that by Corollary. 10,  $A \subseteq \text{EA}(\sigma_{\exists})$ . We show that  $\sigma_{\exists}$  is strongly winning. Let  $h$  be a history such that there is  $\sigma'_{\exists}$  such that  $\emptyset \neq \text{Out}(\sigma'_{\exists}) \cap h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega} \subseteq (L)$ . We prove that  $\text{Out}(\sigma_{\exists}) \cap h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega} \subseteq L$  which shows the result.

If  $\text{Out}(\sigma_{\exists}) \cap h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega} = \emptyset$  then the property is obviously satisfied. Otherwise there is  $w \in \text{Out}(\sigma_{\exists}) \cap h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega}$ .

Consider the strategy  $\sigma_{\exists}[h \leftarrow \sigma'_{\exists}]$  that plays according to  $\sigma_{\exists}$  and when  $h$  is reached shifts to  $\sigma'_{\exists}$ . Formally, given a history  $h'$ :

$$\sigma_{\exists}[h \leftarrow \sigma'_{\exists}] = \begin{cases} \sigma'_{\exists}(h') & \text{if } h \text{ is a prefix of } h' \\ \sigma_{\exists}(h') & \text{otherwise} \end{cases}$$

Since  $h$  is compatible with  $\sigma'_{\exists}$  and  $\text{Out}(\sigma'_{\exists}) \cap h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega} \subseteq L$ , we also have  $\sigma_{\exists}[h \leftarrow \sigma'_{\exists}] \cap h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega} \subseteq L$ . Moreover all outcomes that are not in  $h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega}$  are compatible with  $\sigma_{\exists}$ . Hence  $\text{EA}(\sigma_{\exists}) \cup h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega}$  is sufficient for  $\sigma_{\exists}[h \leftarrow \sigma'_{\exists}]$ . By optimality of assumption  $\text{EA}(\sigma_{\exists})$ ,  $\text{EA}(\sigma_{\exists}) \not\subseteq \text{EA}(\sigma_{\exists}) \cup h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega}$ . Hence  $h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega} \subseteq \text{EA}(\sigma_{\exists})$ . Since  $\text{EA}(\sigma_{\exists})$  is sufficient for  $\sigma_{\exists}$ ,  $\text{Out}(\sigma_{\exists}) \cap h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega} \subseteq L$ , which shows the result.  $\square$

**Example 14.** Consider the specification  $L = (i_1 \Rightarrow \mathbf{X}o_1 \wedge i_2 \Rightarrow \mathbf{X}o_2)\mathbf{U}o_2$ , for which a Büchi automaton is given in Figure 7. There is no winning strategy in this game, since if the input is always  $i_1$  then controller must reply by  $o_1$ , by the first member of the conjunction, and this prevents the third member from being satisfied. However, if the current history is of the form  $(i_1 \cdot o_1)^* \cdot i_2$ , then controller has a winning strategy which consists in replying  $o_2$  and continue to imitate the inputs. Strongly winning strategies must therefore present this behaviour for all  $(i_1 \cdot o_1)^* \cdot i_2$  that are compatible with it.

A strongly winning strategy  $\sigma_{\exists}$  is the one that if the first input is  $i_2$  then it plays the winning strategy we described and otherwise the first input is  $i_1$  and it always play  $o_2$ . This is a strongly winning strategy since for histories beginning with  $i_2$  it is winning and for any other history compatible with  $\sigma_{\exists}$  there is no winning strategy. The assumption corresponding to this strategy is  $\text{EA}(\sigma_{\exists}) = i_2 \cdot \Sigma_O \cdot (\Sigma_I \cdot \Sigma_O)^{\omega} + i_1 \cdot (\Sigma_O \cdot \Sigma_I)^* \cdot o_1 \cdot (\Sigma_I \cdot \Sigma_O)^{\omega}$ . This is indeed an optimal assumption, but it may not be what we would expect because the expression  $i_1 \cdot (\Sigma_O \cdot \Sigma_I)^* \cdot o_1$  is an assumption which talks about the controller rather than the environment. A controller that falsifies the assumption would then be considered correct. Instead of this, we would prefer an assumption which talks only about the environment. This motivates the search for nonrestrictive assumptions.

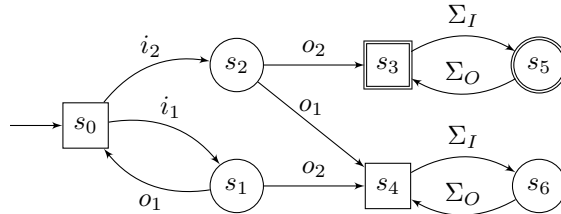


Figure 7: Büchi automaton for specification  $(i_1 \Rightarrow \mathbf{X}o_1 \wedge i_2 \Rightarrow \mathbf{X}o_2)\mathbf{U}o_2$ .

### 3.3 Infinity of optimals

**Theorem 15.** *There is a specification for which there are an infinite number of optimal assumptions and an infinite number of optimal ensurable assumptions.*

*Proof.* Consider the game of Figure 8. In this game there are an infinite number of strongly winning strategies. They must all play  $o_2$  in  $s_5$  but have the choice of how long to stay in  $s_2$ . We write  $\sigma_{\exists}^n$  the strategy that plays  $o_1$ ,  $n$  times before playing  $o_2$  (note that we could also consider strategies that depend on the choice of input in  $s_1$ , but this will not be necessary here). The sufficient hypothesis for  $\sigma_{\exists}^n$  is



We now show that  $\sigma'_\exists[H \setminus W \rightarrow \sigma_\exists]$  is strongly winning for  $L$ . Let  $h$  be a history such that there exists  $\sigma''_\exists$  and  $\emptyset \neq h \cdot (\Sigma_I \cdot \Sigma_O)^\omega \subseteq L$ . Since  $\sigma'_\exists$  is subgame winning, we know that  $\text{Out}_h(\sigma'_\exists) \subseteq L$ . This implies that  $h \in W$ , then  $\sigma'_\exists[H \setminus W \rightarrow \sigma_\exists]$  plays according to  $\sigma'_\exists$  for the rest of the play which means  $\text{Out}_h(\sigma'_\exists[H \setminus W \rightarrow \sigma_\exists]) \subseteq L$ .

Then  $\sigma'_\exists[H \setminus W \rightarrow \sigma_\exists]$  is subgame winning for  $L$  and winning for  $H \Rightarrow L$ . By Thm. 13,  $\text{EA}(\sigma'_\exists[H \setminus W \rightarrow \sigma_\exists])$  is an optimal assumption for  $L$ . Since it is winning for  $H \Rightarrow L$ ,  $\text{Out}(\sigma'_\exists[H \setminus W \rightarrow \sigma_\exists]) \cap H \subseteq L$ . This means that  $\text{EA}(\sigma'_\exists[H \setminus W \rightarrow \sigma_\exists]) = L \cup ((\Sigma_I \cdot \Sigma_O)^\omega \setminus \text{Out}(\sigma'_\exists[H \setminus W \rightarrow \sigma_\exists]))$  contains  $H$ .  $\square$

## 4 Ensurable assumptions

### 4.1 Necessary and sufficient non-restrictive assumptions

In this section, we show properties of assumption that are not restrictive. As we have seen in Lem. 3, this coincide with ensurable assumptions for  $\omega$ -regular objectives.

Given a strategy  $\sigma_\exists$  of Eve, the word  $w$  is *doomed* for  $\sigma_\exists$  if there is an index  $k$  such that one outcome of  $\sigma_\exists$  has prefix  $w_{\leq k}$  and all outcome of  $\sigma_\exists$  that have prefix  $w_{\leq k}$  do not satisfy  $L$ . We write  $\text{Doomed}(\sigma_\exists)$  for the set of words that are doomed for  $\sigma_\exists$  i.e.  $\text{Doomed}(\sigma_\exists) = \{w \mid \exists k \in 2 \cdot \mathbb{N}. \text{Out}(\sigma_\exists) \cap w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \neq \emptyset \text{ and } \text{Out}(\sigma_\exists) \cap w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap L = \emptyset\}$ . We consider the assumption  $\text{EA}^-(\sigma_\exists) = \text{EA}(\sigma_\exists) \setminus \text{Doomed}(\sigma_\exists)$ .

**Lemma 18.** *Let  $\sigma_\exists$  be a strategy, we have the following properties:*

1.  $\text{EA}^-(\sigma_\exists)$  is sufficient for  $\sigma_\exists$ , and nonrestrictive;
2. for all assumption  $A$  sufficient for  $\sigma_\exists$  and not output-restrictive, we have that  $A \subseteq \text{EA}^-(\sigma_\exists)$ .

*Proof.* **1.** The fact that  $\text{EA}^-(\sigma_\exists)$  is sufficient for  $\sigma_\exists$  is a consequence of the fact that it is included in  $\text{EA}(\sigma_\exists)$  which is sufficient for  $\sigma_\exists$  (Lem. 9). Let us show that  $\text{EA}^-(\sigma_\exists)$  is not output-restrictive. Let  $w \in \text{EA}^-(\sigma_\exists)$ , we show that for all  $k$ , and all strategy  $\sigma'_\exists$ , if  $w_{\leq k}$  is prefix of some outcome of  $\sigma'_\exists$  then  $w_{\leq k}$  can be completed in a word of  $\text{EA}^-(\sigma_\exists) \cap \text{Out}(\sigma'_\exists)$ .

Assume that  $w_{\leq k}$  is prefix of an outcome of  $\sigma'_\exists$ . Since  $w$  is not doomed for  $\sigma_\exists$ , either  $w_{\leq k}$  is not a prefix of an outcome of  $\sigma_\exists$ , or there is an outcome  $w'$  of  $\sigma_\exists$  that has prefix  $w_{\leq k}$  and satisfies  $L$ .

- If  $w_{\leq k}$  is not a prefix of an outcome of  $\sigma_\exists$ , then any suffix of  $w_{\leq k}$  belongs to  $\text{EA}^-(\sigma_\exists)$ , by selecting an arbitrary outcome of  $\sigma'_\exists$  after  $w_{\leq k}$  we obtain a word of  $\text{EA}^-(\sigma_\exists) \cap \text{Out}(\sigma'_\exists)$ .
- Otherwise let  $w'$  outcome of  $\sigma_\exists$  that has prefix  $w_{\leq k}$  and satisfies  $L$ . The word  $w'$  is not doomed: any prefix of  $w'$  is prefix of a word (in particular  $w'$ ) which is an outcome of  $\sigma_\exists$  and satisfy  $L$ . Let  $\sigma_\forall$  be a strategy of the environment such that  $\text{Out}(\sigma_\exists, \sigma_\forall) = w'$ . Let  $w'' = \text{Out}(\sigma'_\exists, \sigma_\forall)$ .
  - If  $w'' = w'$ . The word  $w'$  belongs to  $\text{EA}(\sigma_\exists)$  and it is not doomed, hence  $w''$  belongs to  $\text{EA}^-(\sigma_\exists)$ .
  - Otherwise, we have that  $w'' \notin \text{Out}(\sigma_\exists)$  and therefore  $w'' \in \text{EA}(\sigma_\exists)$ . Lets show that  $w''$  is not doomed. Let  $k$  be the smallest index such that  $w''_k \neq w'_k$  and let  $k'$  be another index. If  $k' < k$  then  $w''_{\leq k'} < w'_{\leq k'}$  and since  $w'$  is not doomed, there is an outcome of  $\sigma_\exists$  that have prefix  $w''_{\leq k'}$  and satisfy  $L$ . If  $k' \geq k$  then  $w''_{\leq k'}$  is prefix of no outcome of  $\sigma_\exists$ . Therefore  $w''$  is not doomed for  $\sigma_\exists$  and  $w'' \in \text{EA}^-(\sigma_\exists)$ .

This shows that  $\text{EA}^-(\sigma_\exists)$  is not output-restrictive.

**2.** Lets now assume that  $A$  is sufficient for  $\sigma_\exists$  and not output-restrictive. Let  $w \in A$ , since  $A$  is sufficient we have that  $w \in \text{EA}(\sigma_\exists)$  by (2). Let us show that  $w \notin \text{Doomed}(\sigma_\exists)$ .

Towards a contradiction assume  $w \in \text{Doomed}(\sigma_\exists)$  and let  $k$  be the first index such that for all outcome  $w'$  of  $\sigma_\exists$  that have prefix  $w_{\leq k}$ ,  $w' \not\models L$ . Since  $A$  is not output restrictive,  $A \cap w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \neq \emptyset$ , and  $\text{Out}(\sigma_\exists) \cap w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \neq \emptyset$ , we have that  $A \cap \text{Out}(\sigma_\exists) \cap w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \neq \emptyset$ . Therefore there is a word  $w'$  that has prefix  $w_{\leq k}$  and belongs to  $A \cap \text{Out}(\sigma_\exists)$ . By definition of  $w_{\leq k}$ , the word  $w'$  does not satisfy  $L$ . This contradicts the fact that  $A$  is sufficient for  $L$ .

This proves that  $w$  is not doomed for  $\sigma_{\exists}$  and therefore belongs to  $\text{EA}^-(\sigma_{\exists})$  which shows that  $A \subseteq \text{EA}^-(\sigma_{\exists})$ .  $\square$

**Example 19.** For the strategy  $\sigma_{\exists}$  we defined in example 14, the set of doomed histories is  $i_1 \cdot o_2 \cdot (\Sigma_I \cdot \Sigma_O)^\omega$ . Then  $\text{EA}^-(\sigma_{\exists})$  is  $i_2 \cdot \Sigma_O \cdot (\Sigma_I \cdot \Sigma_O)^\omega$  which is nonrestrictive. This assumption describes better than  $\text{EA}(\sigma_{\exists})$  the assumptions on the environment necessary to win. However it is not optimal among nonrestrictive assumptions, and we will now characterise the strategies for which  $\text{EA}^-(\sigma_{\exists})$  is optimal.

## 4.2 Link with non-dominated strategies

We use the notion of weak dominance classical in game theory. Intuitively a strategy dominates another one if it performs at least as well against any strategy of the environment.

**Definition 20** ([3]). Strategy  $\sigma_{\exists}$  is *very weakly dominated* from history  $h$  by strategy  $\sigma'_{\exists}$  if for all strategy  $\sigma_{\forall}$  of the environment,  $\text{Out}_h(\sigma_{\exists}, \sigma_{\forall}) \in L \Rightarrow \text{Out}_h(\sigma'_{\exists}, \sigma_{\forall}) \in L$ . It is *weakly dominated* from  $h$  by  $\sigma'_{\exists}$  if moreover  $\sigma'_{\exists}$  is not very weakly dominated by  $\sigma_{\exists}$  from  $h$ . A strategy is said *non-dominated* if there is no strategy that weakly-dominates it from the empty history  $\varepsilon$ . A strategy is *non-subgame-dominated* if there is no strategy that weakly-dominates it from any history  $h$ .

We draw a link between optimal assumptions and non-dominated strategies.

**Lemma 21.** *If  $\text{EA}^-(\sigma_{\exists}) \subseteq \text{EA}^-(\sigma'_{\exists})$  then  $\sigma_{\exists}$  is very weakly dominated by  $\sigma'_{\exists}$ .*

*Proof.* Let  $\sigma_{\forall}$  be a strategy of the environment. Let  $w = \text{Out}(\sigma_{\exists}, \sigma_{\forall})$  and  $w' = \text{Out}(\sigma'_{\exists}, \sigma_{\forall})$ . We aim at showing  $w \models L \Rightarrow w' \models L$ . If  $w \not\models L$  or  $w = w'$  the implication is obvious. Assume now  $w \models L$  and  $w \neq w'$ , we will prove that  $w' \models L$ . Since  $w \neq w'$ , let  $k$  be the greatest index such that  $w_{\leq k} = w'_{\leq k}$ . Note that  $w_k$  is not controlled by the environment, since the  $w$  and  $w'$  result from the same strategy of the environment and  $w_{k+1} \neq w'_{k+1}$ , therefore it is controlled by **Eve**.

Assume towards a contradiction that  $w'$  does not satisfy  $L$ . Then  $w' \in \text{Out}(\sigma'_{\exists}) \setminus L$  so  $w' \notin \text{EA}(\sigma'_{\exists})$  and  $w' \notin \text{EA}^-(\sigma'_{\exists})$ . Using the hypothesis that  $\text{EA}^-(\sigma_{\exists}) \subseteq \text{EA}^-(\sigma'_{\exists})$  this means that  $w' \notin \text{EA}^-(\sigma_{\exists})$ . Therefore either  $w' \in \text{Out}(\sigma_{\exists}) \setminus L$  or  $w'$  is doomed for  $\sigma_{\exists}$ .

- If  $w' \in \text{Out}(\sigma_{\exists}) \setminus L$ , the case is solved as for the case  $\text{EA}(\sigma_{\exists}) \subseteq \text{EA}(\sigma'_{\exists})$ .
- Otherwise  $w'$  is doomed for  $\sigma_{\exists}$ . Let  $k'$  be such that  $w'_{\leq k'}$  is an outcome of  $\sigma_{\exists}$  and all outcome of  $\sigma_{\exists}$  that have prefix  $w_{\leq k}$  do not satisfy  $L$ . If  $k' \leq k$  then since  $w$  has  $w'_{\leq k'}$  for prefix and is an outcome of  $\sigma_{\exists}$ , it does satisfy  $L$  which is a contradiction. Otherwise  $k' > k$ . Since we mentioned previously that  $w_k$  is controlled by **Eve**, the fact that  $w'_{\leq k'}$  is prefix of an outcome of  $\sigma_{\exists}$  means that  $\sigma_{\exists}(w_{\leq k}) = w'_{k+1}$  which contradicts the fact that  $k$  is the greatest index such that  $w_{\leq k} = w'_{\leq k}$ .

Therefore  $w'$  satisfies  $L$  and the implication holds.  $\square$

**Example 22.** Consider the game in Figure 9. There are two kind strategies:  $\sigma_{\exists}$  which starts by playing  $o_1$  and  $\sigma'_{\exists}$  which plays  $o_2$ . We have that  $\sigma'_{\exists}$  weakly dominates  $\sigma_{\exists}$ . The assumption  $\text{EA}(\sigma_{\exists})$  necessary to  $\sigma_{\exists}$  is  $\Sigma_I \cdot o_2 \cdot (\Sigma_I \cdot \Sigma_O)^\omega$  while the assumption necessary to  $\sigma'_{\exists}$  is  $\text{EA}(\sigma'_{\exists}) = \Sigma_I \cdot \Sigma_O \cdot i_1 \cdot (\Sigma_I \cdot \Sigma_O)^\omega$ .  $\text{Doomed}(\sigma'_{\exists}) = \Sigma_I \cdot o_2 \cdot i_2 \cdot \Sigma_O \cdot (\Sigma_I \cdot \Sigma_O)^\omega$  while  $\text{Doomed}(\sigma_{\exists}) = (\Sigma_I \cdot \Sigma_O)^\omega$ . So while  $\text{EA}(\sigma_{\exists})$  and  $\text{EA}(\sigma'_{\exists})$  are incomparable, we indeed have  $\text{EA}^-(\sigma_{\exists}) \subset \text{EA}^-(\sigma'_{\exists})$ .

**Lemma 23.** *If  $\sigma_{\exists}$  is very weakly dominated by  $\sigma'_{\exists}$  then  $\text{Doomed}(\sigma'_{\exists}) \subseteq \text{Doomed}(\sigma_{\exists})$ .*

*Proof.* Let  $w$  be a word doomed for  $\sigma'_{\exists}$ . If  $w$  is not an outcome of  $\sigma_{\exists}$ , consider the first index  $k$  such that  $\sigma_{\exists}(w_{\leq k}) \neq \sigma'_{\exists}(w_{\leq k})$ . Let  $d$  be the index from which  $w$  is doomed, that is  $w_{\leq d}$  is prefix of an outcome of  $\sigma'_{\exists}$  and all outcome that have  $w_{\leq d}$  as prefix do not satisfy  $L$ . Let  $\sigma_{\forall}$  be a strategy of the environment. Note that if  $\text{Out}(\sigma'_{\exists}, \sigma_{\forall})$  has prefix  $w_{\leq d}$  then  $\text{Out}(\sigma_{\exists}, \sigma_{\forall})$  has prefix  $w_{\leq \min(d, k)}$ . Moreover if  $\text{Out}(\sigma'_{\exists}, \sigma_{\forall})$  has prefix  $w_{\leq d}$  then it does not satisfy  $L$  and since  $\sigma'_{\exists}$  very weakly dominates  $\sigma_{\exists}$ ,  $\text{Out}(\sigma_{\exists}, \sigma_{\forall})$  does not satisfy  $L$ . Therefore this means  $w_{\leq d}$  is prefix of an outcome of  $\sigma_{\exists}$  and all outcomes of  $\sigma_{\exists}$  that have this prefix do not satisfy  $L$ . Therefore  $w$  is doomed for  $\sigma_{\exists}$ .  $\square$

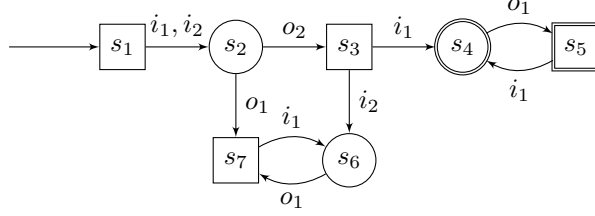


Figure 9: A illustration of a game where **Doomed** makes a difference between restrictive and non-restrictive assumptions.

**Lemma 24.** *If  $\sigma_{\exists}$  is very weakly dominated by  $\sigma'_{\exists}$  then  $EA^{-}(\sigma_{\exists}) \subseteq EA^{-}(\sigma'_{\exists})$ .*

*Proof.* Let  $w \in EA^{-}(\sigma_{\exists}) = EA(\sigma_{\exists}) \setminus \text{Doomed}(\sigma_{\exists})$ . Since  $\sigma_{\exists}$  is very weakly dominated by  $\sigma'_{\exists}$ , we have by Lem. 23 that  $\text{Doomed}(\sigma'_{\exists}) \subseteq \text{Doomed}(\sigma_{\exists})$ . Therefore  $w \notin \text{Doomed}(\sigma'_{\exists})$ .

Assume towards a contradiction that  $w \notin EA(\sigma'_{\exists})$ . Then  $w \notin L$  and  $w \in \text{Out}(\sigma'_{\exists})$ . Let  $w_{\leq k}$  be the longest such that there is an outcome of  $\sigma_{\exists}$  that has this prefix. Note that  $k$  is odd, because  $\sigma_{\exists}$  does not take the decision at odd positions. Since  $w \notin \text{Doomed}(\sigma_{\exists})$  there is an outcome  $w'$  of  $\sigma_{\exists}$  that has  $w_{\leq k}$  for prefix and that is winning. We define the strategy  $\sigma_{\forall}$  of the environment such that:

$$\sigma_{\forall}(h) = \begin{cases} w_{i+1} & \text{if } h = w_{\leq i} \text{ for some } i \\ w'_{i+1} & \text{if } h = w'_{\leq i} \text{ for some } i \\ \sigma'_{\forall}(h) & \text{otherwise} \end{cases}$$

where  $\sigma'_{\forall}$  is some arbitrary strategy that we have fixed. Note that this is well defined because if  $w_{\leq i} = w'_{\leq i}$  and  $i$  is even then  $i \leq k$  and therefore  $w_{i+1} = w'_{i+1}$ . We have that  $\text{Out}(\sigma_{\exists}, \sigma_{\forall}) = w'$  and  $\text{Out}(\sigma'_{\exists}, \sigma_{\forall}) = w$ . Thus  $\text{Out}(\sigma_{\exists}, \sigma_{\forall}) \in L$  and  $\text{Out}(\sigma'_{\exists}, \sigma_{\forall}) \notin L$  which contradicts the fact that  $\sigma_{\exists}$  is very weakly dominated by  $\sigma'_{\exists}$ . Hence  $w \in EA(\sigma'_{\exists}) \setminus \text{Doomed}(\sigma'_{\exists}) = EA^{-}(\sigma'_{\exists})$ .  $\square$

**Theorem 25.** *Let  $L$  be an  $\omega$ -regular specification. If  $\sigma_{\exists}$  is a non-dominated strategy for  $L$ , then  $EA^{-}(\sigma_{\exists})$  is ensurable optimal for  $L$ . Reciprocally if  $A$  is an ensurable optimal assumption for  $L$ , then there is  $\sigma_{\exists}$  a non-dominated strategy for  $L$  such that  $A = EA^{-}(\sigma_{\exists})$ .*

*Proof.*  $\Rightarrow$  Let  $\sigma_{\exists}$  be a non-dominated strategy. Note already that by Lem. 18, the assumption  $EA^{-}(\sigma_{\exists})$  is sufficient for  $L$  and not output-restrictive. We now prove it is optimal. Let  $A$  be a nonrestrictive assumption sufficient for  $L$ . There is a strategy  $\sigma'_{\exists}$  such that  $A$  is sufficient for  $\sigma'_{\exists}$ . By Lem. 9,  $A \subseteq EA(\sigma'_{\exists})$ . As  $\sigma_{\exists}$  is not weakly dominated by  $\sigma'_{\exists}$ , either:

- $\sigma_{\exists}$  is not very weakly dominated by  $\sigma'_{\exists}$ . Then by Lem. 21  $EA^{-}(\sigma_{\exists}) \not\subseteq EA^{-}(\sigma'_{\exists})$ .
- or  $\sigma_{\exists}$  very weakly dominates  $\sigma'_{\exists}$  then by Lem. 24,  $EA^{-}(\sigma'_{\exists}) \subseteq EA^{-}(\sigma_{\exists})$ .

Therefore  $EA^{-}(\sigma_{\exists}) \not\subseteq EA^{-}(\sigma'_{\exists})$  and as  $A \subseteq EA^{-}(\sigma'_{\exists})$ ,  $EA^{-}(\sigma_{\exists}) \not\subseteq A$ . This shows that  $EA^{-}(\sigma_{\exists})$  is optimal among nonrestrictive assumptions for  $L$ .

$\Leftarrow$  Let  $\sigma_{\exists}$  be a strategy such that  $EA^{-}(\sigma_{\exists})$  is nonrestrictive-optimal, we show that  $\sigma_{\exists}$  is non-dominated. Let  $\sigma'_{\exists}$  be a strategy which very weakly dominates  $\sigma_{\exists}$ , we prove that  $\sigma_{\exists}$  very weakly dominates  $\sigma'_{\exists}$ , which shows that  $\sigma_{\exists}$  is not weakly dominated. By Lem. 24,  $EA^{-}(\sigma_{\exists}) \subseteq EA^{-}(\sigma'_{\exists})$ . Since  $EA^{-}(\sigma_{\exists})$  is optimal,  $EA^{-}(\sigma_{\exists}) \not\subseteq EA^{-}(\sigma'_{\exists})$ . Therefore  $EA^{-}(\sigma_{\exists}) = EA^{-}(\sigma'_{\exists})$ . By Lem. 21 this implies that  $\sigma'_{\exists}$  is very weakly dominated by  $\sigma_{\exists}$  which shows the property.  $\square$

A strategy is said *dominant* if it very weakly dominates all strategies. We obtain the following from the previous theorem.

**Corollary 26.** *If  $\sigma_{\exists}$  is dominant then  $EA^{-}(\sigma_{\exists})$  is the unique nonrestrictive optimal assumption.*

*Proof.* A strategy that is dominant is non-dominated and therefore the fact that  $\text{EA}^-(\sigma_\exists)$  is nonrestrictive optimal is a consequence of Thm. 25. To prove uniqueness assume  $A$  is a nonrestrictive optimal assumption. By Thm. 25, there is  $\sigma'_\exists$  such that  $A = \text{EA}^-(\sigma'_\exists)$ . Since  $\sigma_\exists$  is dominant,  $\sigma'_\exists$  is very weakly dominated by  $\sigma_\exists$ . By Lem. 24,  $\text{EA}^-(\sigma'_\exists) \subseteq \text{EA}^-(\sigma_\exists)$  and since  $A$  is optimal we also have  $\text{EA}^-(\sigma_\exists) \subseteq \text{EA}^-(\sigma'_\exists)$  which proves equality and thus uniqueness of the optimal.  $\square$

### 4.3 Computation of optimal ensurable assumptions

In parity games, deciding the existence of a winning strategy from a certain state can be done  $\text{NP} \cap \text{coNP}$  [7]. We will show that if we have available an algorithm for solving parity games then the remaining of the operations to obtain optimal assumptions can be performed efficiently. We first construct a representation of one arbitrary non-dominated strategy. Our construction is based on the notion of memoryless strategies: given  $L$  as a parity automaton, a strategy is said *memoryless* if it only depends on the current state of the automaton, in other words it can be implemented with a Moore machine which has the same structure than the given automaton.

**Lemma 27.** *Given a parity automaton and a memoryless strategy  $\sigma_\exists$  which ensures we are winning from each state in the winning region, we can compute in polynomial time a Moore machine implementing a memoryless non-dominated strategy  $\sigma'_\exists$ .*

*Proof.* By the characterisation of Berwanger [1, Lem. 9], a non-dominated strategy  $\sigma'_\exists$  has to be such that for all history  $h$ , the value of  $h$  is equal to the value of  $\sigma'_\exists$  for  $h$ , for parity games this means:

1. if  $h$  ends with a winning state, then  $\sigma'_\exists$  should be winning from  $h$ ;
2. if  $h$  ends in a state from which there is no winning path,  $\sigma'_\exists$  can behave arbitrarily;
3. if  $h$  ends in another state, then there should exist one outcome of  $\sigma'_\exists$  from  $h$  which is winning.

We define  $\sigma'_\exists$  to behave like  $\sigma_\exists$  in winning states, which already ensures that condition 1 and 2 are satisfied. For states which are not in the winning region but from which there is a winning path, we consider now that all states are controllable and compute a memoryless winning strategy. Such a strategy exists because from all these states there exists a winning path, and it can be computed in polynomial time because there is only one player. We now set  $\sigma'_\exists$  to behave like this strategy on states which are not in the winning region. It ensures that one of its outcome is winning. Hence satisfies all the condition. The corresponding Moore machine is obtained by removing from  $\mathbb{G}$  the transitions that are not taken by  $\sigma'_\exists$ .  $\square$

By combining this construction with a parity automaton for  $L$  it is possible to compute  $\text{EA}(\sigma'_\exists) = L \cup ((\Sigma_I \cdot \Sigma_O)^\omega \setminus \text{Out}(\sigma_\exists))$ .

**Lemma 28.** *Given a Moore machine representing a strategy  $\sigma'_\exists$ , we can construct in polynomial time a safety automaton recognising  $\text{Out}(\sigma_\exists)$ .*

*Proof.* To obtain this automaton we add to the Moore machine states of the form  $S \times \Sigma_I$  and a rejecting state  $\perp$  which is absorbing. The transition function  $\delta'$  is such that for all  $s \in S$  and  $i \in \Sigma_I$ ,  $\delta'(s, i) = (s, i)$ ; if  $G(s, i) = o$  and  $\delta(s, i) = s'$  then  $\delta'((s, i), o) = s'$ ; if  $G(s, i) \neq o$  then  $\delta'((s, i), o) = \perp$ . This defines a safety automaton which recognises  $\text{Out}(\sigma_\exists)$ .  $\square$

**Lemma 29.** *Given  $L$  as a parity automaton and a strategy  $\sigma'_\exists$ , we can compute in polynomial time a parity automaton that recognises  $\text{EA}(\sigma'_\exists)$ .*

*Proof.* By Lem. 28, we can construct a safety automaton that recognises  $\text{Out}(\sigma_\exists)$ . Consider the product between the automaton for  $L$  and for  $\text{Out}(\sigma_\exists)$ . We write  $(s, t)$  for a state of this automaton where  $s$  corresponds to the state for  $L$  and  $t$  for  $\text{Out}(\sigma_\exists)$ . We set the colours to be the same than  $s$  if the colour for  $t$  is 0 and the colour is 0 otherwise. A word that is not in  $\text{Out}(\sigma_\exists)$  will reach  $\perp$  (which has now colour 0) at some point and therefore is accepted by our construction. A word that is in  $L$  is also accepting. Hence this automaton recognises  $\text{EA}(\sigma'_\exists)$ .  $\square$

**Lemma 30.** *If  $\sigma_{\exists}$  is non-subgame-dominated then  $\sigma_{\exists}$  is subgame winning.*

*Proof.* Let  $h$  be a history such that there is a strategy  $\sigma'_{\exists}$  such that  $\text{Out}_h(\sigma'_{\exists}) \subseteq L$ . We will prove that  $\text{Out}_h(\sigma_{\exists}) \subseteq L$  which shows that  $\sigma_{\exists}$  is strongly winning.

Assume  $w \in \text{Out}_h(\sigma_{\exists})$ , and let  $\sigma_{\forall}$  be such that  $w = \text{Out}_h(\sigma_{\exists}, \sigma_{\forall})$ . Consider the strategy  $\sigma_{\exists}[h \leftarrow \sigma'_{\exists}]$  that plays according to  $\sigma_{\exists}$  and when  $h$  is reached shifts to  $\sigma'_{\exists}$ . Formally, given a history  $h'$ :

$$\sigma_{\exists}[h \leftarrow \sigma'_{\exists}] = \begin{cases} \sigma'_{\exists}(h') & \text{if } h \text{ is a prefix of } h' \\ \sigma_{\exists}(h') & \text{otherwise} \end{cases}$$

Note that we already used this construction in the proof of Thm. 13.

Strategy  $\sigma_{\exists}$  is weakly dominated by  $\sigma_{\exists}[h \leftarrow \sigma'_{\exists}]$  from  $h$  because the second strategy is winning from  $h$ . Then, since  $\sigma_{\exists}$  is non-subgame-dominated, it weakly dominates  $\sigma_{\exists}[h \leftarrow \sigma'_{\exists}]$  from  $h$ . We have that  $\text{Out}_h(\sigma'_{\exists}, \sigma_{\forall})$  is winning, therefore  $\sigma_{\exists}[h \leftarrow \sigma'_{\exists}]$  is winning against  $\sigma_{\forall}$ . Since  $\sigma_{\exists}$  weakly dominates  $\sigma_{\exists}[h \leftarrow \sigma'_{\exists}]$ , it is also winning against  $\sigma_{\forall}$  and  $w \in L$ .  $\square$

As strongly non-dominated strategy is also strongly winning (Lem. 30), Lem. 29 already gives an algorithm to compute a (not necessarily ensurable) optimal assumption.

**Corollary 31.** *Given an oracle to compute a memoryless winning strategy in a parity game, we can compute efficiently an optimal assumption as a disjunction of parity automata.*

Now to remove doomed histories, we remove transitions going to states from which there is no winning path.

**Lemma 32.** *Given a parity automaton and a strategy  $\sigma_{\exists}$ , we can compute in polynomial time a safety automaton recognising  $(\Sigma_I \cdot \Sigma_O)^{\omega} \setminus \text{Doomed}(\sigma_{\exists})$ .*

*Proof.* Consider the product between the parity automaton and the Moore machine implementing  $\sigma_{\exists}$ . We consider  $D$  the set of states from which there is no winning path. This set can be computed in polynomial time (it can be seen as the winning states of a one player parity game). To obtain the desired safety automaton, we replace transitions leading to state in  $D$  to a state  $\perp$  that is rejecting.

Let  $w \in \text{Doomed}(\sigma_{\exists})$ , there is a prefix  $h$  of  $w$  such that  $h \cdot (\Sigma_I \cdot \Sigma_O)^{\omega} \cap \text{Out}(\sigma_{\exists}) \cap L = \emptyset$ . Consider the state  $s$  reached of the product reached after reading  $h$ . It is such that no path will be accepting, therefore  $s \in D$ , and  $w$  is rejected by the safety automaton. Similarly if a word is rejected by the automaton, the corresponding path in the product reaches a state in  $D$ , which means it is doomed for  $\sigma_{\exists}$ .  $\square$

Thanks to this lemma, we can easily exclude doomed histories from the previous construction and recognise  $\text{EA}^{-}(\sigma_{\exists})$  which is an ensurable optimal assumption.

**Lemma 33.** *Given a specification as a parity automaton, and a strategy  $\sigma_{\exists}$  as a Moore machine, we can compute in polynomial time a parity automaton recognising  $\text{EA}^{-}(\sigma_{\exists})$ .*

*Proof.* We can construct a parity automaton which recognises  $\text{EA}(\sigma_{\exists})$  by Lem. 29. By Lem. 32, we can construct a safety automaton which recognises  $(\Sigma_I \cdot \Sigma_O)^{\omega} \setminus \text{Doomed}(\sigma_{\exists})$ . We consider the product of these two automata and set the colours to be the same than for  $\text{EA}(\sigma_{\exists})$  except if we reach  $\perp$  in the second component, in which case the colour is 1. A word is accepted if, and only if, it belongs to the intersection of the two languages and therefore to  $\text{EA}^{-}(\sigma_{\exists})$ .  $\square$

**Theorem 34.** *Given a specification as a parity automaton, we can compute in exponential time a parity automaton of polynomial size recognising an ensurable optimal assumption. Moreover, if we have access to an oracle for computing memoryless winning strategies in parity games, our algorithm works in polynomial time.*

*Proof.* We first need to obtain a Moore machine for a memoryless winning strategy  $\sigma_{\exists}$ , this can be done in exponential time or constant time if we have an oracle for that. Then by Lem. 27, we can compute a Moore machine implementing a memoryless non-dominated strategy  $\sigma'_{\exists}$ . By Lem. 33, we can construct a parity automaton recognising  $\text{EA}^{-}(\sigma'_{\exists})$ . By Lem. 25, the language of this automaton is an ensurable optimal assumption.  $\square$

## 4.4 Scenarios

**Theorem 35.** *Let  $L$  be a specification, and  $S$  a coherent scenario compatible with  $L$ . If  $\sigma_{\exists}$  is a non-subgame-dominated, then  $\text{EA}^-([S \rightarrow \sigma_{\exists}])$  is  $\mathcal{E}$ -optimal for  $L$  and  $S$ .*

*Proof.* By Thm. 16 and since  $\sigma_{\exists}$  is subgame winning (Lem. 30),  $\text{EA}([S \rightarrow \sigma_{\exists}])$  is optimal for  $L$  and  $S$ . By Lem. 18,  $\text{EA}^-([S \rightarrow \sigma_{\exists}])$  is ensurable. Assume there are  $A \subseteq (\Sigma_I \cdot \Sigma_O)^\omega$  and  $\sigma'_{\exists}$  such that  $\text{EA}^-([S \rightarrow \sigma_{\exists}]) \subseteq A$ ,  $A$  is ensurable and  $A$  sufficient for  $\sigma'_{\exists}$ . Assume towards a contradiction that there is  $w \in A \setminus \text{EA}^-([S \rightarrow \sigma_{\exists}])$ .

Since  $w \notin \text{EA}^-([S \rightarrow \sigma_{\exists}])$  it is a losing outcome of  $[S \rightarrow \sigma_{\exists}]$  or  $w \in \text{Doomed}([S \rightarrow \sigma_{\exists}])$ . We show that in both cases there is  $i$  such that  $w_{\leq i}$  is not prefix of a word in  $S$  and there is a losing word  $w''$  in  $\text{Out}_{w_{\leq i}}(\sigma'_{\exists})$ :

- If  $w$  is a losing outcome of  $[S \rightarrow \sigma_{\exists}]$ , let  $\sigma_{\forall}$  be such that  $w = \text{Out}([S \rightarrow \sigma_{\exists}], \sigma_{\forall})$ . Let then  $w' = \text{Out}(\sigma'_{\exists}, \sigma_{\forall})$ . Since  $w$  losing and in  $A$  which is sufficient for  $\sigma'_{\exists}$ ,  $w$  is not an outcome of  $\sigma'_{\exists}$  and we have that  $w \neq w'$ . Let  $i$  be the first index such that  $[S \rightarrow \sigma_{\exists}](w_{\leq i}) \neq \sigma'_{\exists}(w_{\leq i})$ . Since both strategies are compatible with  $S$ ,  $w_{\leq i}$  is not a prefix of a word in  $S$ . Therefore by definition of  $[S \rightarrow \sigma_{\exists}]$ ,  $w = \text{Out}_{w_{\leq i}}(\sigma_{\exists})$ . Since  $\sigma_{\exists}$  is non-subgame-dominated and  $w$  losing, there exists a losing outcome  $w''$  in  $\text{Out}_{w_{\leq i}}(\sigma'_{\exists})$ .
- Otherwise there is  $w_{\leq k}$  such that  $w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap \text{Out}([S \rightarrow \sigma_{\exists}]) \neq \emptyset$  and  $w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap \text{Out}([S \rightarrow \sigma_{\exists}]) \cap L = \emptyset$ . Assume towards a contradiction that  $w_{\leq k}$  is prefix of an outcome in  $S$ . Let  $w'$  be that outcome, we have that  $w' \in \text{Out}([S \rightarrow \sigma_{\exists}]) \cap L$  because it is compatible with  $S$  which is compatible with  $L$ . This is a contradiction with the fact that  $w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap \text{Out}([S \rightarrow \sigma_{\exists}]) \cap L = \emptyset$ .

Now, since  $w_{\leq k}$  is not prefix of an outcome in  $S$ ,  $\text{Out}_{w_{\leq k}}([S \rightarrow \sigma_{\exists}]) = \text{Out}_{w_{\leq k}}(\sigma_{\exists})$ . Since  $\sigma_{\exists}$  is non-subgame-dominated and  $\text{Out}_{w_{\leq k}}(\sigma_{\exists}) \cap L = \emptyset$ , there is no winning outcome from  $w_{\leq k}$  (otherwise a strategy allowing this winning outcome would dominated  $\sigma_{\exists}$  from  $w_{\leq k}$ ). Therefore any word in  $w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap A$  is losing, and since  $A$  is sufficient for  $\sigma'_{\exists}$ ,  $w_{\leq k} \cdot (\Sigma_I \cdot \Sigma_O)^\omega \cap A \cap \text{Out}(\sigma'_{\exists}) = \emptyset$ . Let  $w_{\leq i}$  the greatest prefix of a word in  $A$  compatible with  $\sigma'_{\exists}$ . Since  $\sigma_{\exists}$  is non-subgame-dominated and has a losing outcome from  $w_{\leq i}$ , there exists a losing outcome  $w''$  in  $\text{Out}_{w_{\leq i}}(\sigma'_{\exists})$ .

The assumption  $\text{EA}^-([S \rightarrow \sigma_{\exists}]) \cup w''$  is ensurable (as soon as there are at least two input symbols), and it is sufficient for  $[S \rightarrow \sigma_{\exists}]$ . By optimality of  $\text{EA}^-([S \rightarrow \sigma_{\exists}])$ , it includes  $w''$ . Since  $A$  includes  $\text{EA}^-([S \rightarrow \sigma_{\exists}])$ , it also includes  $w''$ , which contradicts that  $A$  is sufficient for  $\sigma'_{\exists}$ .  $\square$

**Lemma 36.** *Given a parity automaton for a coherent scenario  $S$  and a strategy  $\sigma_{\exists}$  we can compute in polynomial time a Moore machine for  $[S \rightarrow \sigma_{\exists}]$ .*

*Proof.* We consider the Moore machine where the state space is the product of the parity automaton and the Moore machine representing  $\sigma_{\exists}$ . We write  $D$  for the states of the parity automaton for  $S$  from which there is no accepting path. If  $(s_1, s_2)$  is a state of the product then  $\delta''((s_1, s_2), i) = (\delta(s_1, i), \delta'(s_2, i))$ , where  $\delta$  and  $\delta'$  are the transition relation for  $S$  and  $\sigma_{\exists}$  respectively. Moreover if  $s_1 \in D$ , then  $G'((s_1, s_2)) = G(s_2)$ , and otherwise we know that in an output state there is only one  $o$  that leads to a state not in  $D$ , we then set  $G'((s_1, s_2)) = o$ . This Moore machine implements  $[S \rightarrow \sigma_{\exists}]$ .  $\square$

**Theorem 37.** *Given a specification  $L$  and a scenario  $S$  as parity automata, we can compute in exponential time a parity automaton of polynomial size recognising a ensurable optimal assumption for  $L$  and  $S$ .*

*Proof.* We can compute in exponential time a memoryless strategy in parity game and as seen in Lem. 27, we can then compute in polynomial time a memoryless non-dominated strategy  $\sigma'_{\exists}$ . Since it is memoryless and non-dominated, it is in fact non-subgame-dominated (this should be clear from the definition of non-subgame-dominated). Then by Lem. 36, we can compute a Moore machine for  $[S \rightarrow \sigma_{\exists}]$ . By Thm. 35, the corresponding assumption  $\text{EA}^-([S \rightarrow \sigma_{\exists}])$  is ensurable optimal for  $L$  and  $S$ . By Lem 33,  $\text{EA}^-([S \rightarrow \sigma_{\exists}])$  can be computed in polynomial time.  $\square$



## 4.5 Generalisation

We have seen in Lem. 17 that from a winning strategy for  $H \Rightarrow L$  and a strongly winning strategy for  $L$ , we could obtain a strategy  $\sigma_{\exists}$  that has both properties. Furthermore, we can compute  $\sigma'_{\exists}$  that is strongly non-dominated for  $L$  and define a strategy that is both non-dominated for  $L$  and winning for  $H \Rightarrow L$ . We define  $\sigma'_{\exists}[H \rightarrow \sigma_{\exists}]$  to be the function that maps  $h$  to  $\sigma_{\exists}(h)$  if  $h \cdot \sigma_{\exists}(h)$  is a prefix of some  $w \in H$  and maps  $h$  to  $\sigma'_{\exists}(h)$  otherwise.

**Lemma 38.** *If  $\sigma_{\exists}$  is winning for  $H \Rightarrow L$  and strongly winning for  $L$  and  $\sigma'_{\exists}$  is strongly non-dominated for  $L$ , then  $\text{EA}^-(\sigma'_{\exists}[H \rightarrow \sigma_{\exists}])$  contains  $H$  and is ensurable-optimal for  $L$ .*

*Proof.* First notice that  $\sigma'_{\exists}[H \rightarrow \sigma_{\exists}]$  is winning for  $H \Rightarrow L$ , since by its definition any of its outcome which belongs to  $H$  is an outcome of  $\sigma_{\exists}$  and thus satisfies  $L$ . We therefore have that  $\text{EA}^-(\sigma'_{\exists}[H \rightarrow \sigma_{\exists}])$  contains  $H$ .

We now show that  $\sigma'_{\exists}[H \rightarrow \sigma_{\exists}]$  is non-dominated for  $L$ . Let  $\sigma''_{\exists}$  be a strategy and  $\sigma_{\forall}$  be an environment strategy such that  $w = \text{Out}(\sigma''_{\exists}, \sigma_{\forall})$  is in  $L$ . Let  $w' = \text{Out}(\sigma'_{\exists}[H \rightarrow \sigma_{\exists}], \sigma_{\forall})$ . Let  $j$  be the largest index such that  $w'_{\leq j} = w_{\leq j}$ . Notice that since the environment strategy is the same for both outcomes, the first time they differ is on an output of the system, hence  $j$  is odd.

If  $w_{\leq j+1}$  is a prefix of a word in  $H$ , then we will follow  $\sigma_{\exists}$  which is strongly winning for  $L$ . If there is a winning strategy from  $w_{\leq j}$  then as  $\sigma_{\exists}$  is strongly winning for  $L$ ,  $w$  satisfies  $L$ . Otherwise there is a strategy  $\sigma'_{\forall}$  compatible with  $w_{\leq j}$  such that  $\text{Out}(\sigma''_{\exists}, \sigma'_{\forall})$  is losing and follows a path of  $H$  after  $w_{\leq j+1}$ . As  $\sigma_{\exists}$  is winning for  $H \Rightarrow L$ ,  $\text{Out}(\sigma'_{\exists}[H \rightarrow \sigma_{\exists}], \sigma'_{\forall}) \in L$ .

If  $w_{\leq j+1}$  is not a prefix of a word in  $H$ , then we play according to  $\sigma'_{\exists}$  which is strongly non-dominated. Hence either  $\text{Out}(\sigma'_{\exists}, \sigma_{\forall}) \in L$  and  $\text{Out}(\sigma'_{\exists}[H \rightarrow \sigma_{\exists}], \sigma_{\forall}) \in L$  or there exists  $\sigma'_{\forall}$  that is compatible with  $w_{\leq j}$  and such that  $\text{Out}(\sigma'_{\exists}, \sigma'_{\forall}) \in L$  and  $\text{Out}(\sigma''_{\exists}, \sigma_{\forall}) \notin L$ , then since it is compatible with  $w_{\leq j}$ ,  $\text{Out}(\sigma'_{\exists}[H \rightarrow \sigma_{\exists}], \sigma'_{\forall}) = \text{Out}(\sigma'_{\exists}, \sigma'_{\forall}) \in L$ . This shows that  $\sigma'_{\exists}[H \rightarrow \sigma_{\exists}]$  is non-dominated.

Then by Thm. 25,  $\text{EA}^-(\sigma'_{\exists}[H \rightarrow \sigma_{\exists}])$  is ensurable-optimal for  $L$ .  $\square$

**Lemma 39.** *Given strategies  $\sigma'_{\exists}$ ,  $\sigma_{\exists}$  as Moore machines and assumption  $H$  as a parity automaton, can construct in polynomial time a Moore machine for  $\sigma'_{\exists}[H \rightarrow \sigma_{\exists}]$ .*

*Proof.* We write  $D$  the set of states of the parity automaton from which there is no accepting path; this can be computed in polynomial time. We write  $\delta'$  and  $\delta$  the transition function for the Moore machine of  $\sigma'_{\exists}$  and  $\sigma_{\exists}$  respectively,  $G'$  and  $G$  their output functions and  $\delta''$  for transition of the parity automaton recognising  $H$ . We consider a Moore machines whose state space is the product of the two Moore machines, the parity automaton and a fourth component in  $\{1, 2\}$  to mark which strategy to follow; we write  $(s, t, u, v)$  a state of the product (note that  $s$  corresponds to  $\sigma'_{\exists}$  and  $t$  to  $\sigma_{\exists}$ ). We now describe the transition function  $\delta'''$  of this Moore machine for a state  $(s, t, u, v)$  and an input  $i$ , and the output function  $G''$ . If  $v = 1$  we follow  $\sigma_{\exists}(h)$  then  $G''(s, t, u, v) = G(t)$  and otherwise  $v = 2$ , we follow  $\sigma'_{\exists}(h)$  and  $G''(s, t, u, v) = G'(s)$ . If  $u$  is an input state and  $\delta''(u, G(\delta(t, i))) \notin D$  then the path following  $\sigma_{\exists}$  is prefix of some word in  $H$ , so  $\sigma'_{\exists}[H \rightarrow \sigma_{\exists}]$  follows  $\sigma_{\exists}(h)$ , which means for us  $\delta'''((s, t, u, v), i) = (\delta'(s, i), \delta(t, i), \delta''(u, i), 1)$ . Otherwise we follow  $\sigma'_{\exists}(h)$ , which means  $\delta'''((s, t, u, v), i) = (\delta'(s, i), \delta(t, i), \delta''(u, i), 2)$ . This implements the strategy  $\sigma'_{\exists}[H \rightarrow \sigma_{\exists}]$ .  $\square$

**Theorem 40.** *There is an exponential algorithm that given  $L$  and  $H$  sufficient for  $L$  as parity automata, computes a parity automaton whose language  $H'$  is such that  $H \subseteq H'$  and  $H'$  ensurable-optimal for  $L$ .*

*Proof.* Assume we are given automata  $\mathcal{A}_H$  for  $H$ , and  $\mathcal{A}_L$  for  $L$ . We construct  $\mathcal{A}_{H \Rightarrow L}$  recognising  $L \cup (\Sigma_I \cdot \Sigma_O)^{\omega} \setminus H$ . We first compute a winning strategy  $\sigma_{\exists}$  in  $\mathcal{A}_{H \Rightarrow L}$ . We can compute in exponential time a memoryless strategy  $\sigma'_{\exists}$  which is winning in  $\mathcal{A}_L$  from all states from which there is a winning strategy [11], it is in fact strongly winning. By Lem. 39 we can construct a Moore machine for  $\sigma'_{\exists}[H \rightarrow \sigma_{\exists}]$ , and by Lem. 33 we can obtain an automaton for  $\text{EA}^-(\sigma'_{\exists}[H \rightarrow \sigma_{\exists}])$ . Thanks to Lem. 38,  $\text{EA}^-(\sigma'_{\exists}[H \rightarrow \sigma_{\exists}])$  is ensurable optimal for  $L$  and contains  $H$ .  $\square$

## 5 Input-assumptions

### 5.1 Infinity of optimals

As we show now, in general there can be an infinite number of incomparable assumptions that are sufficient.

**Theorem 41.** *There is a specification  $L$  for which there are an infinite number of optimal input assumptions.*

*Proof.* Consider the Büchi automaton in Figure 6 with  $\Sigma_I = \{i_1, i_2\}$  and  $\Sigma_O = \{o_1, o_2\}$ , and  $L$  its language. In the automaton, the objective of the controller is to reach  $s_3$ . For that, it can only influence the transition out of state  $s_1$ . Given  $n$ , we define the language  $A_n$  to be  $S^\omega \setminus \{(\Sigma_I \cdot \Sigma_O)^n \cdot i_1 \cdot \Sigma_O \cdot (\Sigma_I \cdot \Sigma_O)^\omega\}$ . Under assumption  $A_n$ , the winning strategy  $\sigma_\exists$  consists in playing  $o_2$  at step  $n$ . If assumption  $A_n$  is respected the path goes to  $s_3$  and is accepting. This shows that  $A_n$  is sufficient for  $L$ .

Each assumption  $A_n$  is a safety assumption and an input assumption: it is safety because all words not in  $A_n$  have a prefix in  $(\Sigma_I \cdot \Sigma_O)^n \cdot i_1$  which is a bad prefix, and it is an input assumption because if  $w \in A_n \wedge \pi_I(w) = \pi_I(w')$  then  $w'_{2n+1} = w_{2n+1} \neq i_1$  and therefore  $w' \in A_n$ .

We now show that each  $A_n$  is optimal. Let  $B$  be a sufficient assumption such that  $A_n \subseteq B$  and  $\sigma_\exists$  be a strategy such that  $\text{Out}(\sigma_\exists) \cap B \subseteq L$ . Let  $w \notin A_n$  we will prove  $w \notin B$ , which shows  $B \subseteq A_n$  and optimality of  $A_n$ . By definition of  $A_n$ ,  $w_{2n+1} = i_1$ . Let  $w' = \text{Out}(\sigma_\exists, \pi_I(w))$ , we have that  $\pi_I(w') = \pi_I(w)$  and since  $B$  is an input assumption  $w' \in B \Leftrightarrow w \in B$ . Assume towards a contradiction that  $w' \in L$ . Then there is an index  $k \geq 1$  such that  $w'_{2k} = o_2$ ,  $w'_{2k+1} = i_2$  and for all  $1 \leq j < k$ ,  $w'_{2j} = o_1$ . Since  $w_{2n+1} = i_1$ ,  $k \neq n$ .

- If  $k > n$  then let  $w'' = \text{Out}(\sigma_\exists, u)$  where  $u = \pi_I(w')_{\leq n} \cdot i_2 \cdot i_1^\omega$ . We have  $w''_{2n+1} = i_2$  therefore  $w'' \in A_n \subseteq B$ , and  $w'' \in \text{Out}(\sigma_\exists)$ . After reading  $w''_{\leq 2n} = w'_{\leq 2n}$  we are still in state  $s_0$ . After that we read  $i_2$  which brings us in  $s_1$  and then the inputs are all  $i_1$ , therefore  $w''$  is losing which contradicts  $B \cap \text{Out}(\sigma_\exists) \subseteq L$ .
- If  $k < n$  then let  $w'' = \text{Out}(\sigma_\exists, u)$  where  $u_{k+1} = i_1$ ,  $u_{n+1} = i_2$  and  $u_j = \pi_I(w')_j$  otherwise. We have that  $w'' \in A_n \subseteq B$ , and  $w'' \in \text{Out}(\sigma_\exists)$ . Since  $k < n$ ,  $w''_{\leq 2k} = w'_{\leq 2k}$  and  $w''_{2k+1} = i_1$ , therefore  $w''$  is losing which contradicts  $B \cap \text{Out}(\sigma_\exists) \subseteq L$ .

This shows that  $w' \in \text{Out}(\sigma_\exists) \setminus L$  and since  $\text{Out}(\sigma_\exists) \cap B \subseteq L$ ,  $w' \notin B$  and as a consequence  $w \notin B$ .  $\square$

### 5.2 Link with remorsefree strategies

In this section we show a link between input assumptions and a class of strategies called remorsefree.

**Definition 42** (Remorsefree). Given a specification  $L$ , a strategy  $\sigma_\exists$  is *remorsefree* if for all  $\sigma'_\exists$  and  $w \in \Sigma_I^\omega$ ,  $\text{Out}(\sigma'_\exists, w) \models L$  implies  $\text{Out}(\sigma_\exists, w) \models L$ . This is the notion used in [6] for dominance. A strategy  $\sigma_\exists$  is *remorsefree-admissible* if for all  $\sigma'_\exists$  either  $\forall w \in \Sigma_I^\omega$ .  $\text{Out}(\sigma'_\exists, w) \models L \Rightarrow \text{Out}(\sigma_\exists, w) \models L$  or  $\exists w \in \Sigma_I^\omega$ .  $\text{Out}(\sigma_\exists, w) \models L \not\models \text{Out}(\sigma'_\exists, w) \models L$ .

**Lemma 43.** *Given a strategy  $\sigma_\exists$  of Eve, if  $L \neq \emptyset$  then the assumption*

$$IA(\sigma_\exists) = \{w \in (\Sigma_I \cdot \Sigma_O)^\omega \mid \pi_I(w) \notin \pi_I(\text{Out}(\sigma_\exists) \setminus L)\}$$

*is an input-assumption that is sufficient for  $\sigma_\exists$ . Moreover if  $A$  is an input-assumption which is sufficient for  $\sigma_\exists$  then  $A \subseteq IA(\sigma_\exists)$ .*

*Proof.* First notice that  $IA$  is an input assumption: if  $\pi(w) = \pi(w')$  then  $w \in IA(\sigma_\exists) \Leftrightarrow w' \in IA(\sigma_\exists)$ .

We now show that  $IA(\sigma_\exists)$  is sufficient for  $\sigma_\exists$ . Let  $w \in IA(\sigma_\exists) \cap \text{Out}(\sigma_\exists)$ . Since  $w \in IA(\sigma_\exists)$ ,  $\pi_I(w) \notin \pi_I(\text{Out}(\sigma_\exists) \setminus L)$ . Therefore  $w \notin \text{Out}(\sigma_\exists) \setminus L$ . Hence  $w \in L$ .

Finally, we show that if  $A$  is an input-assumption sufficient for  $\sigma_\exists$  then  $A \subseteq IA(\sigma_\exists)$ . Let  $A$  be a sufficient input-assumption and  $w \in A$ . Assume towards a contradiction that  $w \notin IA(\sigma_\exists)$ . Then  $\pi_I(w) \in \pi_I(\text{Out}(\sigma_\exists) \setminus L)$ . Let  $w' \in \text{Out}(\sigma_\exists) \setminus L$  such that  $\pi_I(w') = \pi_I(w)$ . Since  $A$  is sufficient,  $w' \notin A$ . Since  $A$  is an input-assumption and  $\pi_I(w') = \pi_I(w)$  we also have  $w \notin A$  which is a contradiction.  $\square$

In [6], Finkbeiner and al. show the following.

**Theorem 44.** [6] *There is a remorsefree strategy if, and only if, there is a unique minimal assumption for  $L$ . Moreover the minimal assumption is the input-assumption sufficient for the remorsefree strategy.*

**Example 45.** Consider the game of Figure 10. In this game the only remorsefree strategy is to output  $o_1$  at the first step. The corresponding assumption is  $A = \Sigma_I \cdot \Sigma_O \cdot (\{i_2, i_3\} \cdot \Sigma_O)^\omega$  while the assumption corresponding to the strategy outputting  $o_2$  is  $\Sigma_I \cdot \Sigma_O \cdot (\{i_3\} \cdot \Sigma_O)^\omega$  which is more restrictive. The assumption  $A$  is indeed the unique optimal input-assumption.

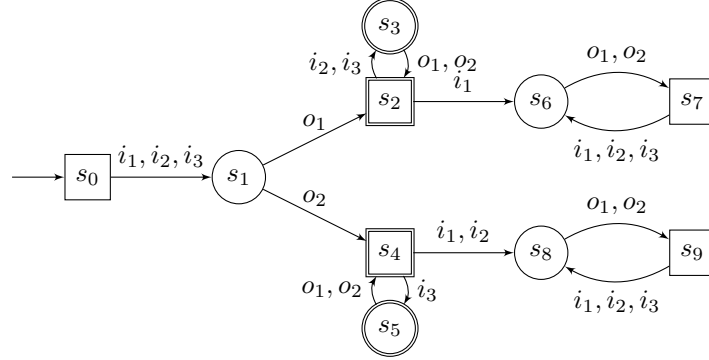


Figure 10: Büchi automaton for which the remorsefree strategy consists in outputting  $o_1$ .

We now use the associated notion of admissibility to characterise the minimal assumptions that are sufficient to win.

**Theorem 46.** *If  $\sigma_\exists$  is a remorsefree admissible strategy for  $L$ , then  $IA(\sigma_\exists)$  is an optimal input-assumption for  $L$ . Reciprocally if  $A$  is an optimal input-assumption for  $L$ , then there is a remorsefree admissible strategy  $\sigma_\exists$ , such that  $A = IA(\sigma_\exists)$ .*

*Proof.*  $\Rightarrow$  Let  $\sigma_\exists$  be a remorsefree-admissible strategy and  $A$  the corresponding environment assumption. Let  $B$  be such that  $A \subset B$ . We show that  $B$  is not sufficient for  $L$  which will show that  $A$  is optimal.

Let  $\sigma'_\exists$  be a strategy we show that  $B$  is not sufficient for this strategy. Since  $\sigma_\exists$  is remorsefree-admissible, one of those two cases occurs:

- $\forall w' \in \Sigma_I^\omega$ .  $\text{Out}(\sigma'_\exists, w') \models L \Rightarrow \text{Out}(\sigma_\exists, w') \models L$ . Let  $w \in B \setminus A$ . Since  $A = IA(\sigma_\exists)$ , we have that  $w \in \pi_I(\text{Out}(\sigma_\exists) \setminus L)$ . Hence  $\text{Out}(\sigma_\exists, w) \not\models L$ , and we have that  $\text{Out}(\sigma'_\exists, w) \not\models L$  which shows that  $B$  is not sufficient for  $\sigma'_\exists$ .
- or  $\exists w' \in \Sigma_I^\omega$ .  $\text{Out}(\sigma_\exists, w') \models L \wedge \text{Out}(\sigma'_\exists, w') \not\models L$ . We have that  $w'$  belongs to  $A$  by definition, thus it belongs to  $B$  by hypothesis, and since  $\text{Out}(\sigma'_\exists, w') \not\models L$ ,  $B$  is not sufficient for  $\sigma'_\exists$ .

$\Leftarrow$  Let  $A$  be an optimal assumption for  $L$  and  $\sigma_\exists$  the corresponding strategy. We will show that  $\sigma_\exists$  is remorsefree-admissible. For that, let  $\sigma'_\exists$  be another strategy of **Eve**, we show that either:  $\forall w \in \Sigma_I^\omega$ .  $\text{Out}(\sigma'_\exists, w) \models L \Rightarrow \text{Out}(\sigma_\exists, w) \models L$  or  $\exists w \in \Sigma_I^\omega$ .  $\text{Out}(\sigma_\exists, w) \models L \not\models \text{Out}(\sigma'_\exists, w) \models L$ .

Since  $A$  is optimal, this means that it is not strictly included in  $B = IA(\sigma'_\exists)$ . This means that either  $A = B$  or  $A \setminus B \neq \emptyset$ .

- if  $A = B$  then we show that  $\forall w \in \Sigma_I^\omega$ .  $\text{Out}(\sigma'_\exists, w) \models L \Rightarrow \text{Out}(\sigma_\exists, w) \models L$ :
  - if  $w \in A = IA(\sigma_\exists)$ , we have that  $\text{Out}(\sigma_\exists, w) \models L$ , and the implication holds.
  - if  $w \notin A = B = IA(\sigma'_\exists)$ , we have that  $\text{Out}(\sigma'_\exists, w) \not\models L$ , and the implication holds.
- Otherwise  $A \setminus B \neq \emptyset$  then let  $w \in A \setminus B$ . Since  $w \in A$ ,  $\text{Out}(\sigma_\exists, w) \models L$  and since  $w \notin B = IA(\sigma'_\exists)$ ,  $\text{Out}(\sigma'_\exists, w) \not\models L$ . This shows that  $\text{Out}(\sigma_\exists, w) \models L \not\models \text{Out}(\sigma'_\exists, w) \models L$ .

□

## References

- [1] D. Berwanger. Admissibility in infinite games. In *Proc. of STACS'07*, volume 4393 of *LNCS*, pages 188–199. Springer, Feb. 2007.
- [2] R. Bloem, R. Ehlers, S. Jacobs, and R. Könighofer. How to handle assumptions in synthesis. In *Proceedings 3rd Workshop on Synthesis, SYNT 2014, Vienna, Austria, July 23-24, 2014.*, pages 34–50, 2014.
- [3] R. Brenguier, J. Raskin, and M. Sassolas. The complexity of admissibility in omega-regular games. In *CSL-LICS '14, 2014*. ACM, 2014.
- [4] R. Brenguier, J.-F. Raskin, and O. Sankur. Assume-admissible synthesis. In L. Aceto and D. de Frutos-Escrig, editors, *CONCUR*, volume 42 of *LIPICs*, pages 100–113. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.
- [5] K. Chatterjee, T. A. Henzinger, and B. Jobstmann. Environment assumptions for synthesis. In *CONCUR 2008-Concurrency Theory*, pages 147–161. Springer, 2008.
- [6] W. Damm and B. Finkbeiner. Does it pay to extend the perimeter of a world model? In *FM 2011: Formal Methods*, pages 12–26. Springer, 2011.
- [7] E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy. In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, pages 368–377. IEEE, 1991.
- [8] M. Faella. Admissible strategies in infinite games over graphs. In *MFCS 2009*, volume 5734 of *Lecture Notes in Computer Science*, pages 307–318. Springer, 2009.
- [9] R. Faran and O. Kupferman. Spanning the spectrum from safety to liveness. In *Automated Technology for Verification and Analysis*, pages 183–200. Springer, 2015.
- [10] E. Filiot, N. Jin, and J.-F. Raskin. An antichain algorithm for LTL realizability. In *Computer Aided Verification*, pages 263–277. Springer, 2009.
- [11] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata Logics, and Infinite Games: A Guide to Current Research*. Springer-Verlag New York, Inc., New York, NY, USA, 2002.
- [12] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 179–190. ACM, 1989.